



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

El Problema de Rutas de Vehículos Con Capacidad

Autor/es

ÁNGELA RUIZ-NAVARRO SÁENZ

Director/es

FRANCISCO JAVIER PÉREZ LÁZARO

Facultad

Facultad de Ciencia y Tecnología

Titulación

Grado en Matemáticas

Departamento

MATEMÁTICAS Y COMPUTACIÓN

Curso académico

2018-19



El Problema de Rutas de Vehículos Con Capacidad, de ÁNGELA RUIZ-
NAVARRO SÁENZ

(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported. Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los titulares del copyright.



UNIVERSIDAD DE LA RIOJA

Facultad de Ciencia y Tecnología

TRABAJO FIN DE GRADO

Grado en Matemáticas

El Problema de Rutas de Vehículos Con Capacidad

Realizado por:

Ángela Ruiz Navarro Sáenz

Tutelado por:

Francisco Javier Pérez Lázaro

Logroño, Junio, 2019

Resumen

El Problema de Rutas de Vehículos con Capacidad.

En este documento se estudiará el problema de rutas de vehículos con capacidad (CVRP). Este problema consiste en, dados una serie de clientes, cada uno de ellos con una ubicación y demanda de cierto producto específico, y una flota de vehículos, todos con la misma ubicación y capacidad, encontrar la ruta mínima que deben de seguir los vehículos en la que se visiten todos los clientes una única vez, se satisfagan las demandas de todos los clientes y los vehículos vuelvan a su punto de partida sin haber excedido su capacidad.

Plantearemos un modelo en términos de programación lineal y se analizarán dos métodos para la resolución de estos problemas. El primero de ellos se trata del primer algoritmo que se publicó para solucionar dichos problemas (1959), un algoritmo heurístico planteado por Dantzig y Ramser; y el segundo es un algoritmo de ramificación y acotación exacto, publicado en 1986 a manos de Laporte, Mercure y Nobert.

Abstract

The Capacited Vehicle Routing Problem.

This document will study the capacited vehicle routing problem (CVRP). This problem consist of a set of customers, each with a known location and a known requirement for a basic product, which must be supplied from a single deposit by delivery vehicles of known capacity. The problem is to design the minimum routes for these vehicles, subject to the following conditions and restrictions: the requirements of all customers must be met, customers must be visited only once by a vehicle, the fleet of vehicles must begin and end in the depot and the capacity of the vehicles should not be violated.

We will propose a linear programming model and two methods for solving these problems will be analyzed. The first one is the first algorithm that was published to solve these problems (1959), and it is an heuristic algorithm proposed by Dantzig and Ramser. And the second one is an exact branch and bound algorithm, published in 1986 by Laporte, Mercure and Nobert.

Índice general

1. Introducción.	5
2. Variantes del VRP.	11
3. Modelos de programación lineal	15
3.1. El problema del viajante	15
3.2. El problema de rutas de vehículos	16
3.2.1. El problema de rutas de vehículos capacitado	17
4. The Truck Dispatching Problem	21
4.1. El problema de reparto de gasolina	21
4.1.1. Introducción	21
4.1.2. Problema	21
4.1.3. Procedimiento Computacional	24
4.2. Consideraciones acerca del artículo	33
5. Un algoritmo de ramificación y acotación.	35
5.1. Descripción	35
5.2. Formulación	35
5.3. Algoritmo	37
5.4. Regla de partición	38
5.5. Ejemplo	40
Bibliografía	49
Anexo	50

Capítulo 1

Introducción.

En este trabajo se estudiará el problema de rutas de vehículo con capacidad (CVRP, “Capacited Vehicle Routing Problem”), una de las variantes más conocidas del problema de rutas de vehículos, VRP. Dicho problema consiste en planificar las rutas que deben de seguir una flota de vehículos, encargados de transportar unas ciertas cantidades de un producto específico para cubrir las demandas de ciertos clientes, todos ellos con ubicaciones distintas conocidas. La flota de vehículos posee una capacidad específica y se encuentra en una ubicación determinada (depósito) de donde deberán partir. El objetivo de este problema es encontrar la ruta mínima que deben de seguir estos vehículos teniendo en cuenta una serie de restricciones.

- Los vehículos deben partir y terminar en el depósito.
- Las demandas de los clientes deben ser cubiertas.
- Cada cliente solo puede ser visitado una única vez.
- La carga que llevan los vehículos no pueden superar su capacidad.

Este problema es una generalización del conocido Problema del Viajante (TSP, “Travelling Salesman Problem”), en el cual se busca la ruta mínima que debe seguir un trabajador, en la que visite a todos los clientes una única vez y vuelva a su punto de partida.

Estos dos problemas pertenecen al campo de la optimización combinatoria, y son problemas NP-duros.

Son problemas muy presentes en la vida cotidiana, es por ello que se les ha dado una gran importancia, y han sido multitud de autores los que los han estudiado durante una amplia etapa de la vida.

Una referencia importante en este tema la encontramos en el siglo XIX en Alemania, en el libro “Der Handlungsreisende Von einem alten Commis-Voyageur”, que describe la necesidad de encontrar tours buenos (eficientes) para un viajante de comercio. Este libro no lo plantea desde un punto de vista matemático, sino que da una serie de consejos y presenta cinco rutas a través de regiones de Alemania y Suiza. Cuatro de ellas, incluyen visitas de regreso a una ciudad anterior que sirve de base para esa parte del viaje. Y la quinta es una ruta del problema del viajante. El autor llega a obtener resultados casi óptimos, o quizás óptimos para las carreteras que había entonces.

Aunque la primera vez que se escribió desde el punto de vista matemático acerca del problema del viajante fue en 1920 (Viena) y lo hizo Karl Menger. Menger era economista y matemático centrado en la geometría hiperbólica y probabilística y el álgebra y uno de sus temas favoritos fue el estudio de técnicas para medir longitudes de curvas en el espacio. Estas investigaciones dieron lugar a un problema muy parecido al problema del viajante, que en sus inicios se denominó el Problema del Cartero (“The Messenger Problem”), un problema que definió la tarea de encontrar, para un número finito de puntos cuyas distancias entre pares eran conocidas, la ruta más corta que pudiese seguir el cartero y visitase todos los puntos, pero sin volver a su lugar de inicio.

El matemático Merrill M. Flood (1937) trató de obtener soluciones casi óptimas en referencia a las distribuciones de autobuses escolares en Estados Unidos. Tanto Flood como A. W. Tucker recuerdan que escucharon el problema por primera vez en un seminario de Hassler Whitney en Princeton en 1934. Flood fue quien más tarde (1948) hizo popular dicho problema como “El Problema de los 48 Estados de Hassler Whitney”, quien se cree que lo hizo conocido en la Corporación RAND, la cual ofreció un premio monetario.

Julia Robinson resolvió otro de los problemas de la corporación RAND, pero no obtuvo recompensa por ser una empleada, probablemente inspirada por otro problema de la corporación, comenzó su estudio en el TSP en 1949.

No se sabe exactamente cuándo fue la primera vez que se denominó dicho problema como el Problema del Viajante, pero se cree que fue a manos de Merrill Flood, aunque la primera referencia descubierta es a manos de Robinson en un trabajo de investigación de 1949, que lo usa de forma espontánea, lo que sugiere que en ese concepto fuese un término familiar. Robinson formula el problema en el cual un vendedor comienza su ruta en Washington, visita todas las capitales estatales y regresa a Washington, que coincide con la descripción de Flood y el conjunto de datos empleados por Dantzig. Este problema conecta el TSP y el “Problema de los 48 estados”.

Era tal la importancia de dicho problema que a la vez pero otra parte del mundo (India) el estadístico Mahalanobis (1940) realizó un estudio sobre muestras de la superficie de yute en Bengal, en el que discutió aspectos de las soluciones del TSP a través de ubicaciones elegidas al azar en el plano euclideo para recopilar datos que sirviesen para pronosticar los cultivos. El objetivo de Mahalanobis fue saber más acerca de los recorridos que surgen cuando las ubicaciones de la ciudad se eligen al azar en una unidad cuadrada, es decir cada punto (x,y) con x e y entre 0 y 1, y siendo igualmente probable que se elija una ubicación u otra.

En 1954 Dantzig, Fulkerson y Jonshon, resolvieron un TSP con 48 ciudades [7], en el que plantearon un modelo matemático y un método de planos de corte, que influenció a la mayoría de algoritmos del TSP que se escribieron después, y precedió a Gomory, quien se basó en este método para crear su método clásico de resolución de un problema cualquiera de programación entera mediante planos de corte. Dantzig, Fulkerson y Jonshon publicaron un artículo [15] en 1959 en el cual se resuelve un problema paso a paso con diez ciudades.

Dantzig fue el creador del método del Simplex y está considerado como uno de los funda-

dores de la programación lineal, junto con Neumann y Kantorovich.

Dada la importancia que estaba teniendo el problema del viajante, en 1959 se planteó el problema de rutas de vehículos, la primera vez a manos de Dantzig y Rasmser, que entonces, fue denominado El Problema de Reparto de Gasolina (“The Truck Dispatching Problem”) [6], como una aplicación al mundo real en relación con el suministro de gasolina a las estaciones de servicio. En este artículo los autores propusieron por primera vez un enfoque algorítmico para el CVRP, dicho algoritmo se trataba de un algoritmo heurístico.

Un poco más tarde, en 1964 fueron Clarke y Wright los que propusieron otro algoritmo heurístico para solucionar el CVRP.

Se consideró un problema tan interesante y útil para la vida real, que poco a poco se fueron publicando gran cantidad de artículos en revistas de interés matemático, presentando modelos matemáticos y proponiendo tanto algoritmos exactos como algoritmos heurísticos para encontrar la solución óptima a dicho problema.

Varios de los primeros autores que propusieron algoritmos exactos para la resolución de CVRP fueron Christofides y Eilon (1969) [13] y Laporte, Mercure y Nobert (1986) [14].

Para resolver CVRPs se emplearon métodos exactos y heurísticos, dentro de los exactos encontramos algoritmos de ramificación y acotación, y ramificación y corte. Y dentro de los heurísticos tenemos heurísticas constructivas, heurísticas de mejora clásica, metaheurísticas, hibridaciones y algoritmos unificados.

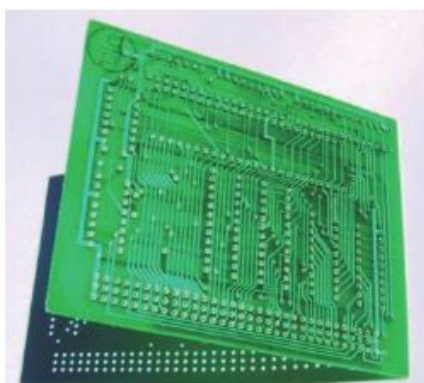
Tal ha sido la importancia del problema de rutas con capacidad (CVRP) que veremos algunas de sus muchas aplicaciones. La principal aplicación del problema de rutas de vehículos con capacidad la encontramos en el ámbito de la logística, en el cual deseamos repartir/entregar una cierta cantidad de un producto a varios clientes, y disponemos de una flota de vehículos, dichos vehículos sólo pueden transportar una cantidad máxima de producto (capacidad del vehículo) y deben partir del depósito, donde se encuentra el producto, visitar a los clientes una única vez y volver al depósito, habiendo recorrido la mínima ruta posible.

Otras de sus muchas aplicaciones son:

- En el ámbito de la logística encontramos varias aplicaciones más, como lo pueden ser los GPSs, ya que algunos softwares de mapas que se ejecutan en el GPS incluyen solucionadores de TSP, que se usan para dar estimaciones de tiempo para viajar de un punto a otro, las recogidas y entregas tanto de personas (autobuses) como de productos (camiones), las comidas a domicilio, la planificación de excursiones y viajes en las que deseamos conocer el mayor número de lugares o monumentos en el menos tiempo posible...
- **Mapeo de Genomas:** otra aplicación muy diferente al movimiento de personas, vehículos o productos surge en la investigación genética. Un mapa del genoma tiene para cada cromosoma una secuencia de marcadores con estimaciones de las distan-

cias entre los marcadores adyacentes, dichos marcadores son segmentos de ADN que aparecen exactamente una vez en el genoma. La capacidad de reconocer estos genomas únicos les permite a los investigadores usarlos para verificar, comparar y combinar mapas físicos creados en diferentes laboratorios. Es importante tener información precisa sobre el orden en el que aparecen los marcadores en el genoma, es aquí donde entra en juego el TSP.

- Otro lugar donde juega un papel importante es en el **tema industrial**, ya que como sabemos, en la fabricación en masa las máquinas tienen el objetivo de realizar una tarea repetidas veces, como puede ser taladrar. Veamos algunos ejemplos:
 - Perforación y soldadura de placas de circuitos: Las placas de circuitos que contienen algunos dispositivos electrónicos suelen tener orificios para montar chips de ordenadores o para hacer conexiones entre capas. Estos agujeros los hacen taladradoras automáticas que se van moviendo entre determinadas ubicaciones para ir creando dichos orificios. Es aquí donde nuestro objetivo es minimizar el tiempo de viaje del cabezal de perforación durante el proceso de producción. Al igual que con el tema de la soldadura, en el que debemos encontrar la manera más rápida de mover el dispensador a través de una serie de puntos para imprimir la pasta de soldadura.



- Latón grabado: Las matrices de latón se emplean para imprimir imágenes con relieve, como las que podemos encontrar en las cajas de bombones. Estas matrices fueron hechas a mano, pero ahora se graban con fresas. Cuando la máquina termina el corte de la letra o símbolo, el eje se eleva y el dispositivo pasa a la siguiente letra. Los elementos a cortar no son puntos únicos por lo que la máquina puede moverse a cualquier ubicación por encima del elemento. En 2008 se probó que la aplicación del TSP a este problema redujo el tiempo de trabajo incluso a la mitad en los casos en que las matrices tienen cantidades significativas de texto o donde hay patrones abstractos de muchos puntos.
- Limpieza de placas de silicio: Los chips estándares están grabados en placas de silicio, estas placas deben estar limpias de impurezas, hay empresas que tienen técnicas para limpiar estas impurezas de las placas, y usan este tipo de problema para guiar la maquinarias de un punto a otro.
- Pruebas para microprocesadores: la conocida firma informática NVIDIA adoptó pruebas posteriores a la fabricación de circuitos de chip utilizadas para probar sus procesadores gráficos. El TSP se utiliza para determinar el orden de los puntos de escaneo para que la cadena sea lo más corta posible, minimizar esta

longitud ayuda a ahorrar espacio de cableado en el chip y tiempo en la fase de prueba al permitir que las señales se envíen con más rapidez.

- Patrones de corte en la industria del vidrio.
- **Organización de datos:** En los tours musicales se emplea para organizar a artistas en orden circular, de manera que, artistas similares estén cerca unos de otros. En esta aplicación las ciudades son los músicos y los gastos son las medidas de similitud.
- **Programación de trabajos:** Las empresas que ofrecen productos de hardware y software para gestionar la producción de tarjetas de crédito, identificación... en las cuales se producen muchos tipos de tarjetas en el mismo hardware, requieren pasos de reconfiguración entre ejecuciones de producción, como lo puede ser el cambio en el color de la cinta, este tiempo de configuración entre diferentes trabajos es significativo y reduce la producción diaria, por lo que emplean el TSP para secuenciar los trabajos en un orden que minimice el tiempo total de configuración. En este caso, las ciudades son los trabajos, y el costo entre los trabajos es el tiempo que se necesita para reconfigurar las máquina para el siguiente trabajo.

A continuación daremos una visión más amplia de lo que son los problemas de rutas de vehículos con capacidad (CVRP).

En el primer capítulo, “Introducción”, hemos planteado el problema en cuestión, conocido cuales han sido los orígenes tanto del problema del viajante como del problema de rutas de vehículos con capacidad, y mostrado algunas de las muchas aplicaciones del problema de rutas de vehículos con capacidad y el del viajante. Para ello nos hemos basado en libro de William J. Cook [2] y el de Toth y Vigo [1].

En el segundo capítulo, “Variantes del VRP”, estudiaremos las variantes del VRP que se han desarrollado a lo largo de los años, para ellos hemos empleado el libro de Toth y Vigo [1].

En el tercer capítulo, “Modelos de programación lineal para el TSP y el CVRP”, se dará una definición más formal del problema del viajante (TSP) [4] y el problema de rutas de vehículos con capacidad (CVRP) [5] y se planteará un modelo de programación lineal para la resolución de cada uno de ellos.

En el cuarto capítulo, “The Truck Dispatching Problem: Primer artículo sobre el CVRP”, se analizará el primer artículo en el que se presentó un algoritmo matemático para la resolución de CVRPs. Este fue publicado en 1959 por Dantzig y Ramser [6]. Se trata de un algoritmo heurístico, es decir, proporcionará una solución aproximada.

Y por último, en el quinto y último capítulo, denominado “Un algoritmo de ramificación y acotación”, como se ya visto un algoritmo heurístico, se analizará un artículo publicado en 1986 por Laporte, Mercure y Nobert [14], el cual describe un algoritmo de ramificación y acotación para la resolución de dichos problemas. Se trata de un algoritmo exacto, es decir, que proporcionará la solución óptima del problema. A continuación se aplicará dicho algoritmo a el ejemplo estudiado en el capítulo 4.

Como el ejemplo que hemos tratado en el capítulo 5 no ha podido desarrollarse al completo, veremos una resolución más detallada en un Anexo. Aquí se muestra paso por paso

la solución al problema que resolvieron Dantzig y Ramser, con el algoritmo propuesto en el capítulo 5. Se comienza aplicando paso a paso el algoritmo (resolución en anchura) hasta un cierto punto. Puesto que la resolución a mano de un problema de estas características es muy costosa, es por ello que después, y dado que se conoce la solución óptima al problema por el artículo “The Truck Dispatching Problem” será guiada (resolución en profundidad) para llegar antes a la solución óptima.

Capítulo 2

Variantes del VRP.

Dado que los problemas de rutas de vehículos han tenido una gran relevancia en la vida real, ha dado lugar a que surgiesen importantes variantes. Estas variantes son:

- **El problema de rutas de vehículos con capacidad:** comúnmente conocido como CVRP por sus siglas en inglés “Capacited Vehicle Routing Problem”, es una de las variantes más importantes del problema de rutas de vehículos, su definición es la misma que la del VRP incluyendo una restricción, los vehículos no pueden transportar más de cierta carga, esta carga es la capacidad del vehículo. El CVRP a su vez, puede subdividirse en el ACVRP Y SCVRP. El primero es el problema de rutas de vehículos con capacidad asimétrico, es decir, la distancia de ida y vuelta entre dos puntos no es la misma, y el segundo es el problema de rutas de vehículos con capacidad simétrico, en el que la distancia de ida y vuelta entre dos puntos es la misma.
- **El problema de rutas de vehículos con ventanas horarias:** comúnmente conocido como VRPTW por sus siglas en inglés “Vehicle Routing Problem with Time Windows”, es otra de las variantes más importantes del VRP. Es una extensión del problema de rutas de vehículos con capacidad, donde cada cliente debe recibir su pedido dentro de un intervalo de tiempo asociado, este intervalo de tiempo se denomina ventana horaria. Las ventanas horarias pueden ser duras o blandas. El caso de las ventanas horarias blandas es aquel en el que se permiten alterar los límites de entrega. El caso en el que el vehículo llegue demasiado pronto a un cliente, este deberá esperar hasta que el cliente esté listo para comenzar el servicio, como norma general este caso no suele conllevar ningún costo adicional. Sin embargo, si fuese el cliente el que tuviese que esperar a que el vehículo llegase, y recibir el pedido fuera de su ventana horaria, esto ocasionaría un costo adicional. En el caso de las ventanas horarias duras no se permite ninguna violación del tiempo de entrega.
- **El problema de recolección y entrega:** conocido como PDP, por sus siglas en inglés “Pickup-and-Delivery Problem”. Los problemas de recolección y entrega constituyen una parte muy importante de los problemas de rutas de vehículos, en los cuales, las mercancías o pasajeros tienen que ser transportados desde diferentes orígenes a diferentes destinos. Los PDPs se pueden clasificar en tres categorías
 - **De muchos a muchos (M-M):** cada producto puede tener múltiples orígenes y destinos y cualquier ubicación puede ser el origen o el destino de múltiples productos. Un ejemplo de este problema es el inventario de tiendas.

- **De uno a muchos y de muchos a uno (1-M-1):** estos problemas se caracterizan por la presencia de algunos productos que se entregarán desde un depósito a muchos clientes y de otros productos que se recogerán en los clientes y se transportarán de vuelta al depósito. Un ejemplo de estos problemas es la recogida de latas y botellas vacías en bares, es decir, desde un depósito se reparten los refrescos a cada cliente, y en cada clientes se recogen las botellas vacías para ser devueltas al depósito.
 - **Uno a uno(1-1):** Por último los problemas uno a uno son aquellos en los que cada producto tiene un único origen y un único destino. Un ejemplo de este problema es la mensajería que recibimos en nuestras casas.
- Dentro de los problemas de ventanas horarias también encontramos los problemas de pasajeros, dichos pasajeros son recogidos cada uno de ellos en una determinada ubicación y entregados en otra ubicación diferente, con unas restricciones de tiempo específicas, a este tipo de problemas se les denomina **Problemas de Dial-a-Ride (DARP)**. Un ejemplo sería un autobús o ambulancia, que va recogiendo a personas mayores, o personas con dificultades, para dejarlas en sus respectivas residencias, hospitales o centros, para que puedan llevar a cabo sus actividades como puede ser rehabilitación. De lo que se trata es de asignar la solicitud de cada cliente a un vehículo y diseñar las rutas para que los vehículos realicen todos los servicios dentro de las restricciones de tiempo con un costo mínimo.
 - **Problemas de rutas de vehículos estocásticos:** (SVRP) por sus siglas en inglés “Stochastic Vehicle Routing Problems”. Hay ocasiones en las que no todos los parámetros que participan en el problema están claros, es decir, puede que haya incertidumbre en algunos de estos parámetros. Esta incertidumbre pueden provenir tanto de variaciones esperadas como inesperadas, que pueden afectar a varios aspectos del problema, como por ejemplo, parámetros estocásticos que conlleven costos adicionales. Esto suele ocurrir en el caso de los modelos e VRP que se usan para planificar operaciones logísticas. Este es el caso en el que en el momento en el que se planifican las rutas el flujo informativo es distinto al que se tiene cuando se ejecutan. Por ejemplo, si las demandas de los clientes no son claras, y se concretan cuando son visitados, las rutas planeadas con anterioridad pueden ser inviables, ya que la demanda total de los clientes puede superar la capacidad del vehículo. En estos casos, se debe tomar costos adicionales que suelen involucrar decisiones adicionales para dar una solución factible.
Se suelen considerar tres tipos de parámetros estocásticos
 1. **Exigencias estocásticas:** los volúmenes de producto que se deben recoger o entregar a los clientes son aleatorios.
 2. **Clientes estocásticos:** los clientes pueden estar presentes o ausentes.
 3. **Tiempos estocásticos:** tanto los tiempos de servicio a los clientes como los tiempos de viaje de los vehículos pueden considerarse estocásticos.
 - **VRP con red de retorno:** VRPB por sus siglas en inglés “VRP with Backhauls”. En este problema los clientes están partidos en dos conjuntos, el primero (linehaul) en el que los clientes requieren la entrega de los productos desde el depósito, y los segundos (backhaul) que tienen productos para ser recogidos y transportados al depósito. En la solución de este problema el servicio combinado de líneas y redes de

retorno en la misma ruta es factible, pero todos los clientes de linehaul reciben su pedido antes de que se recoja la carga de los clientes backhaul.

- **VRP con flota heterogénea o mixta:** HFVRP por sus siglas en inglés “Heterogeneous or mixed fleet VRP”. En este problema la flota de vehículos se compone de distintos vehículos, cada uno con capacidad y costos específicos. La elección del vehículo apropiado para ser asignado a cada ruta puede tener un gran impacto en la solución final, tanto a nivel operativo, en el cual se da la flota heterogénea, como a nivel estratégico, en el cual la composición óptima de la flota tiene que ser determinada.
- **VRP periódico:** PVRP por sus siglas en inglés “Periodic VRP”. Aquí las entregas se hacen en varios días. A cada cliente se le debe asignar el día o los días en los que recibirá su pedido. Y la planificación de la ruta se realizará para cada día. Por lo tanto se debe de tomar tres decisiones; la elección de los días de visita de cada cliente, la agrupación de clientes en visitas por día y la ruta del vehículo para cada agrupación. Un ejemplo de este tipo de problemas, son las rutas que lleva a cabo la conocida empresa Ecoembes.
- **VRP con entrega dividida:** SDVRP por sus siglas en inglés “Split Delivery VRP”. En este tipo de problema los clientes pueden ser atendidos más de una vez dividiendo la cantidad del pedido entre las diferentes visitas. Por lo tanto, se requiere la determinación del número de visitas por cliente, la proporción de demanda que se debe entregar a cada cliente en cada visita y la ruta de los vehículos.
- **Problema de rutas de vehículos con beneficios:** VRPP por sus siglas en inglés “Vehicle Routing Problem with Profits”. En este tipo de problema hay una gran diferencia con los demás, ya que no se da el conjunto de clientes que se va a atender. Por lo que el objetivo del problema es saber los clientes a los que se debe atender; cómo agrupar los clientes de diferentes rutas y cómo ordenar los clientes de cada ruta. Normalmente, se asocia un beneficio a cada cliente, haciéndolo así más o menos atractivo. Es así que el problema se puede medir tanto en términos de costos como de ganancias.
- **Problemas de rutas dinámicas de vehículos:** al resolver problema de rutas de vehículos, se suele disponer de todos los parámetros de entrada, pero esto no es así en la vida real, donde hay parámetros como la demanda del cliente, los tiempos de viaje... que están incompletos o incluso llegan a ser desconocidos. Por lo tanto es muy importante tener en cuenta dos dimensiones al resolver estos problemas. La primera de ellas es la evolución de la información, es decir, si la información está sujeta a cambios, incluso cuando ya se ha planificado la ruta, y la segunda la calidad de los datos, es decir, la viabilidad o el conocimiento de los datos disponibles. Estas dos dimensiones tienen algo en común, la resolución de estas dudas se lleva a cabo a la vez que se ejecuta la ruta.

Capítulo 3

Modelos de programación lineal para el TSP y el CVRP.

3.1. El problema del viajante

El problema del viajante (conocido como TSP, por sus siglas en inglés, “Travelling salesman problem”); consiste en encontrar una ruta para el viajante, la cual salga desde un lugar en concreto, visite una cantidad de ciudades establecidas y vuelva al lugar de inicio, habiendo visitado todas las ciudades una única vez y siendo la distancia recorrida lo mínima posible.

Aunque a simple vista, el recorrido que debe seguir un repartidor, comercial... en su día a día no parezca complicado, en términos matemáticos, el problema de planificación de rutas, suele presentar un problema de optimización combinatoria bastante complejo.

Veamos ahora una definición mas formal [4, pg.3]. Sea $G = (V, A)$ un grafo, $V = \{0, 1, 2, \dots, n\}$ el conjunto de nodos, y $C = (c_{ij})_{n \times n}$ la matriz de costos, donde la entrada c_{ij} corresponde al costo de viajar del nodo i al nodo j .

Dependiendo de la matriz de costos, el problema del viajante puede dividirse en dos tipos. Si C es simétrica, lo que quiere decir que G no está dirigido, hablamos del problema del viajante simétrico (STSP), y si C fuese asimétrica (grafo dirigido) estaríamos en el caso del problema del viajante asimétrico (ATSP).

Aunque el caso del STSP puede verse como un caso especial del ATSP ya que cada grafo no dirigido puede considerarse un grafo dirigido.

Sin embargo, el ATSP también se puede considerar como un caso particular del STSP duplicando los nodos. Consideramos un digrafo $D = (N, E)$, construimos el grafo no dirigido $G^* = (V^*, A^*)$ con el conjunto de nodos $V^* = N \cup \{0^*, 1^*, 2^*, \dots, n^*\}$ donde $N = \{0, 1, 2, \dots, n\}$. Para cada arco (i, j) en D creamos un arco (i, j^*) en G^* con costo c_{ij} , e introducimos arcos ficticios (i, i^*) con costo $-M$, donde M es un número muy grande; teniendo todos los demás costo M . De tal forma, se puede afirmar, que resolver el ATSP en D es equivalente a resolver el STSP en G^* .

El TSP se puede formular con la programación lineal de enteros, siendo $x_{ij} = 1$ si el arco de ir de la ciudad i a la j está incluida en la solución, o 0 en caso contrario. Dicha formulación se puede describir de la siguiente manera:

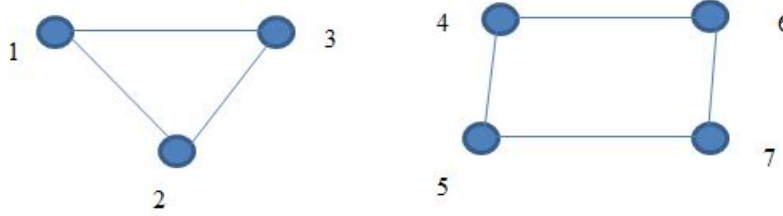
$$\text{Minimizar } \sum_{i=0}^n \sum_{j=0; j \neq i}^n c_{ij} x_{ij} \quad (3.1)$$

Sujeto a:

$$\sum_{i=0}^n x_{ij} = 1 \quad j = 0, 1, \dots, n, \quad (3.2)$$

$$\sum_{j=0}^n x_{ij} = 1 \quad i = 0, 1, \dots, n. \quad (3.3)$$

Con (3.2) se puede asegurar que a cada ciudad j le llegue exactamente una arista. Y con (3.3) que desde cada ciudad se salga exactamente hacia una ciudad. Sin embargo, como se puede ver en el siguiente gráfico, aunque se respeten (3.2) y (3.3) el recorrido obtenido no es un circuito válido para el TSP porque contiene subciclos.



Pero para cualquier solución $X = (x_{ij})_{n \times n}$ del problema anterior, consideramos el conjunto $S \subseteq V$ que representa subconjunto de nodos. Por lo que debemos incluir restricciones adicionales de eliminación de subciclos. Para un subconjunto $S \subseteq V$ se selecciona como máximo $|S| - 1$ arcos.

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad S \neq \emptyset, |S| \leq |V|/2. \quad (3.4)$$

Con esta restricción aseguramos que no puede existir un subgrafo en el cual el número de aristas sea mayor que el número de nodos menos 1, por lo tanto, en cada subconjunto de vértices hay una arista que va a un nodo que no está en dicho subconjunto.

Notar que la ecuación (3.4) también se puede escribir:

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1 \quad S \neq \emptyset, |S| \leq |V|/2. \quad (3.5)$$

3.2. El problema de rutas de vehículos

El problema de rutas de vehículos (VRP, por sus siglas en inglés “Vehicle Routing Problem”) se puede describir como el problema el cual se basa en diseñar una ruta de recolección de pedidos y entrega en varios puntos geográficos distintos sujeto a restricciones.

Este problema desempeña un papel fundamental en el campo de la logística por lo que juega un papel muy importante en la actualidad.

El VRP consiste en diseñar rutas de vehículos con el menor costo posible y que cumplan las siguientes propiedades:

- Cada punto geográfico correspondiente a $V \setminus \{0\}$ es visitado una única vez por sólo un vehículo. En el punto 0 se sitúa el depósito.
- Todas las rutas comienzan y finalizan en el depósito.
- Se cumplen las restricciones descritas en el capítulo 2, dependiendo de la variante del problema.

3.2.1. El problema de rutas de vehículos capacitado

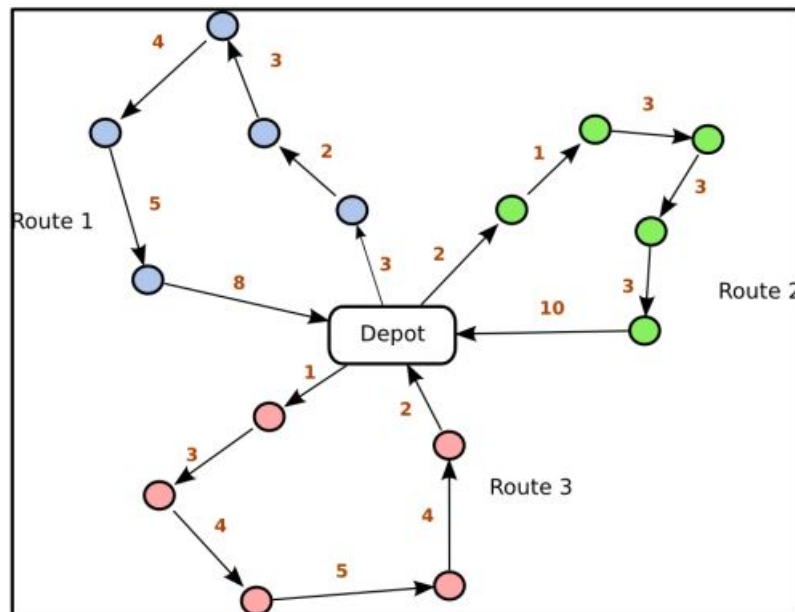
El CVRP puede definirse de la siguiente manera [5]; sea $G = (V, A)$ un grafo completo donde $V = \{0, 1, 2, \dots, n\}$ son los vértices y A el conjunto de aristas/arcos. El vértice 0 corresponde al depósito, y todos los demás a los clientes, cada uno de ellos con una demanda d_j no negativa, mientras que el depósito tendrá una demanda ficticia $d_0 = 0$.

Dado un conjunto de clientes $S \subseteq V$ la demanda total del conjunto está dado por $d(S) = \sum_{j \in S} d_j$.

Un costo no negativo c_{ij} está asociado a cada arco que va del nodo i al j ($(i, j) \in A$) que representa el gasto del viaje que hay de ir del nodo i al j . El caso de arcos en forma de bucle no están permitidos, por lo que al arco (i, i) se le asigna un costo $c_{ii} = +\infty \forall i \in V$.

Si la matriz de costos es asimétrica, G es un grafo dirigido y al problema se le conoce como ACVRP, por el contrario si fuese simétrica hablaríamos de una grafo no dirigido y entonces estaríamos en el caso del SCVRP y el conjunto de arcos A se suele sustituir por el conjunto de arcos no dirigidos E al que se le asocia un costo c_e .

Dado un conjunto de vértices $S \subseteq V$ para el problema simétrico denotamos el conjunto de arcos como $\delta(S) = \{(i, j), i \in S, j \notin S \text{ or } i \notin S, j \in S\}$ y $\sigma(S) = \{(i, j), i, j \in S\}$ que tienen solo uno o ambos puntos finales en S respectivamente.



Normalmente trabajamos con grafos completos (cada vértice está unido con todos los demás vértices), si este no fuese el caso asignaríamos un costo infinito a los arcos inexistentes.

Tenemos K vehículos en el depósito disponibles, todos ellos con una capacidad C , asumiendo que cada vehículo puede usarse en a lo sumo una ruta, siendo $K \geq K_{min}$, con K_{min} el número mínimo de vehículos necesarios para satisfacer la demanda de los clientes.

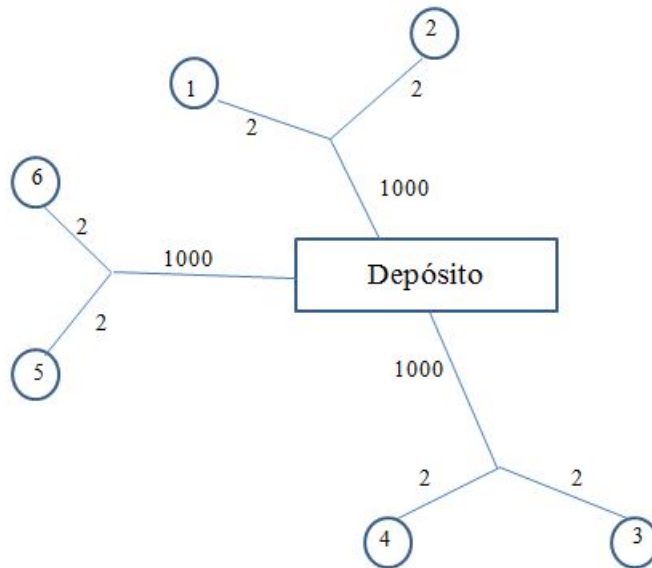
Dado un conjunto $S \subseteq V \setminus \{0\}$, denotamos el número mínimo de vehículos necesarios para satisfacer la demanda de los clientes como $\gamma(S)$. Teniendo en cuenta que $\gamma(V \setminus \{0\}) = K_{min}$. Asumimos que $d_j \leq C$ para cada $j = 1, 2, 3, \dots, n$.

El CVRP está constituido por una serie de K circuitos simples correspondientes a las rutas de los vehículos, con un costo mínimo tales que:

- Cada circuito visita el depósito
- Cada vértice $j \in V \setminus \{0\}$ es visitado por solo un vehículo.
- La suma de las demandas de los vértices visitados en cada circuito no debe exceder la capacidad del vehículo C .

Se sabe que este problema es un caso especial del problema del viajante, para $C \geq d(V)$ y $K = K_{min} = 1$.

Por otro lado, las soluciones que se fuerzan a que se usen exactamente K vehículos con $K > K_{min}$ no suelen conducir al costo mínimo. Pero hay casos en los que no ocurre, veámoslo con el siguiente ejemplo.



Considérense las seis ciudades dadas junto con sus distancias. Sea 3000 unidades la demanda de cada una de las ciudades. Se dispone de dos vehículos (número mínimo de vehículos), con una capacidad de 10000 cada uno de ellos. Si estuviésemos obligados a

emplear exactamente esos dos vehículos recorreríamos una distancia de $1000 + 2 + 2 + 2 + 2 + 1000 + 1000 + 2 + 2 + 1000 = 4012$ y $1000 + 2 + 2 + 2 + 2 + 1000 + 1000 + 2 + 2 + 1000 = 4012$ que en total sumaría una distancia de 8024. Mientras que si pudiésemos usar tres vehículos, el tercero de ellos también con una capacidad de 10000, recorreríamos una distancia de $1000 + 2 + 2 + 2 + 2 + 1000 = 2008$, $1000 + 2 + 2 + 2 + 2 + 1000 = 2008$ y $1000 + 2 + 2 + 2 + 2 + 1000 = 2008$, que en total sumaría 6024. Podemos concluir, por tanto, que usando tres vehículos minimizamos la distancia.

Podemos decir que minimizar la distancia y el número de vehículos suelen ir unidos, pero no siempre. A excepción del caso $K_{min} = 1$, en el que ambos objetivos coinciden como se verá en la imagen de la página 22.

El problema de rutas de vehículos con capacidad asimétrico

Primero examinaremos el ACVRP, es decir, con una matriz de costos asimétrica. El modelo de programación lineal de enteros que vamos a emplear es de variables binarias, que asigna $x_{ij} = 1$ para indicar si un vehículo atraviesa el arco $(i, j) \in A$ en la solución óptima y $x_{ij} = 0$ en caso contrario.

$$\text{Minimizar } \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (3.6)$$

Sujeto a:

$$\sum_{i \in V} x_{ij} = 1 \quad \forall j \in V \setminus \{0\}, \quad (3.7)$$

$$\sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \setminus \{0\}, \quad (3.8)$$

$$\sum_{i \in V} x_{i0} = K, \quad (3.9)$$

$$\sum_{j \in V} x_{0j} = K, \quad (3.10)$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq \gamma(S) \quad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset, \quad (3.11)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V. \quad (3.12)$$

Con (3.7) y (3.8) aseguramos que a cada nodo se llega de solo un nodo y que de cada nodo se salga a solo un nodo. Con (3.9) y (3.10) se impone que desde el depósito salen K vehículos y llegan K vehículos. Las restricciones de capacidad (3.11) con las que aseguramos que de cada subconjunto S , $\gamma(S)$ aristas siempre salgan del subconjunto, que también se pueden dar transformando las restricciones de eliminación de subciclos de la siguiente forma:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - \gamma(S) \quad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset \quad (3.13)$$

que impone que al menos $\gamma(S)$ arcos dejan cada conjunto de vértices S .

Este planteamiento calcula la distancia mínima para exactamente K vehículos, siendo K el número mínimo de vehículos o no. Si $K > K_{min}$ y lo que quisiésemos fuese, en primer lugar minimizar el número de vehículos, y en segundo lugar, para ese número mínimo de vehículos calcular la distancia mínima debemos definir un grafo $\bar{G} = (\bar{V}, \bar{A})$

que se obtiene de G añadiendo $K - K_{min}$ vértices artificiales a V , cada uno de ellos con una demanda $d_j = 0$. Sea $W = \{n + 1, \dots, n + K - K_{min}\}$ el conjunto de estos vértices artificiales, el costo $\overline{c_{ij}}$ de los arcos $(i, j) \in \overline{A}$ viene dado por:

$$\overline{c_{ij}} = \begin{cases} c_{ij} & (i, j \in V) \\ 0 & (i = 0, j \in W) \\ 0 & (i \in W, j = 0) \\ M & (i \in W, j \in V \setminus \{0\}) \\ M & (i \in V \setminus \{0\}, j \in W) \\ M & (i, j \in W) \end{cases} \quad (3.14)$$

donde M es un número positivo muy grande.

Hay que tener en cuenta que este método favorece a que la solución óptima del problema contenga rutas vacías con cada vértice artificial.

Notar que si cambiásemos la primera M por un c_{0j} podríamos calcular la mínima distancia para K vehículos o menos.

El problema de rutas de vehículos con capacidad simétrico

En este caso no es necesario saber en que dirección va cada arista, y para cada arista no dirigida se usa una variable entera x_e con $e \in E$ para indicar cuantas veces se usa la arista en la solución óptima.

En particular, si $e \notin \delta(0)$ entonces $x_e \in \{0, 1\}$, mientras que si $e \in \delta(0)$ entonces $x_e \in \{0, 1, 2\}$. El caso $x_e = 2$ se da cuando un cliente solo tiene que hacer la ruta de ir a una ciudad (j), es decir, hacer la ruta $0 \rightarrow j \rightarrow 0$.

Podemos expresar el problema de la siguiente manera:

$$\text{Minimizar } \sum_{e \in E} c_e x_e \quad (3.15)$$

Sujeto a:

$$\sum_{e \in \delta(i)} x_e = 2 \quad \forall i \in V \setminus \{0\}, \quad (3.16)$$

$$\sum_{e \in \delta(0)} x_e = 2K, \quad (3.17)$$

$$\sum_{e \in \delta(S)} x_e \geq 2\gamma(S) \quad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset, \quad (3.18)$$

$$x_e \in \{0, 1, 2\} \quad \forall e \in \delta(0), \quad (3.19)$$

$$x_e \in \{0, 1\} \quad \forall e \notin \delta(0). \quad (3.20)$$

Las restricciones (3.16) y (3.17) imponen que a cada vértice inciden exactamente dos arcos, y que en el depósito inciden $2K$ arcos.

Las restricciones (3.18) son las de capacidad, que imponen los requisitos de capacidad del vehículo, forzando a que un número suficiente de arcos entren en cada subconjunto de vértices, que al igual que en el caso asimétrico se pueden dar como restricciones de eliminación de subciclos.

$$\sum_{e \in \sigma(S)} x_e \leq |S| - \gamma(S) \quad S \subseteq V \setminus \{0\}, S \neq \emptyset. \quad (3.21)$$

Capítulo 4

The Truck Dispatching Problem: Primer artículo sobre el CVRP.

4.1. El problema de reparto de gasolina

4.1.1. Introducción

El problema de reparto de gasolina (“The Truck Dispatching Problem”), un problema de rutas de vehículos con capacidad, fue el primero para el que se planteó un algoritmo que resolviese problemas de esta índole.

Este algoritmo fue formulado en 1959 por G. B. Dantzig y J. H. Ramser [6], se trata de un algoritmo heurístico, es decir, un algoritmo no exacto, pero capaz de proporcionar soluciones muy cercanas a la óptima.

En este primer algoritmo lo que se intentaba era proporcionar la ruta óptima que debían de seguir una flota de camiones de reparto de gasolina a granel entre un terminal (depósito) y una serie de estaciones de servicio suministradas por el depósito.

Se proporcionan las distancias más cortas entre dos puntos cualesquiera de los incluidos en el problema y la demanda de cada uno de ellos, a excepción del depósito.

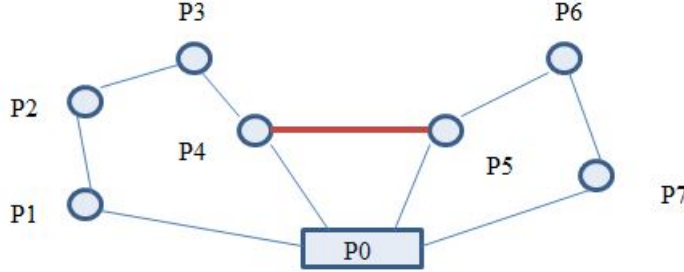
Lo que se intenta es encontrar la forma de asignar estaciones a los camiones de tal forma que se satisfagan las demandas de cada estación y el kilometraje total cubierto por toda la flota sea mínimo. Para ello el algoritmo que se expone a continuación proporciona un procedimiento de programación lineal para obtener una solución casi óptima, estos cálculos podían hacerse tanto a mano como a ordenador. Los autores aseguran que, aunque lo probaron con varios ejemplos y proporcionaba soluciones lo suficientemente próximas a la óptima, no se podía asegurar su buen funcionamiento con todos los problemas.

4.1.2. Problema

El problema del viajante también puede generalizarse imponiendo la condición de que se realicen las d_i entregas específicas (demandas) en cada nodo (excepto para el caso del depósito), a los que denotaremos P_i . Si la capacidad del transportista

$$C \geq \sum_i d_i \tag{4.1}$$

el problema es idéntico al del viajante en su forma original, ya que el transportista puede atender cada punto de entrega en un viaje que enlaza todos los puntos, ya que aunque se dispusiese de más vehículos, la ruta óptima se conseguiría con un solo vehículo debido a la desigualdad triangular. Veámoslo:



Así, como podemos observar en la imagen anterior la ruta óptima sería $P0 \rightarrow P1 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow P5 \rightarrow P6 \rightarrow P7$, en lugar de recorrer $P0 \rightarrow P1 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow P0$ y $P0 \rightarrow P5 \rightarrow P6 \rightarrow P7 \rightarrow P0$.

En el problema de reparto de gasolina la relación entre d_i y C es tal que el transportista solo puede realizar entre 1 y t entregas cada viaje. Por lo tanto, el problema de reparto de gasolina está caracterizado por la siguiente relación

$$C < \sum_i d_i. \quad (4.2)$$

Para simplificar asumiremos que solo se entrega un producto y que todos los camiones tienen la misma capacidad C . El siguiente algoritmo parte de la idea básica de sintetizar la solución en varias etapas de agregación en las que se llevan a cabo suboptimizaciones en pares de puntos o grupos. El número de etapas de agregación que se emplearán se determina de la siguiente manera: se ordenan las entregas d_i en una secuencia d_1, d_2, \dots, d_n de forma que $d_i \leq d_{i+1}$ para cada $i = 1, \dots, n-1$. Después determinamos t tal que

$$\sum_{i=1}^t d_i \leq C \quad \text{y} \quad \sum_{i=1}^{t+1} d_i > C, \quad (4.3)$$

donde t representa el número máximo de entregas que un camión de capacidad C puede realizar para un conjunto dado de d_i 's. Dado que la secuencia d_1, d_2, \dots, d_t representa una combinación factible, puede estar en la solución óptima, el método de cálculo del número de agregaciones a emplear debe admitir la combinación d_1, d_2, \dots, d_t en la agregación final. En cada agrupación se dobla el número de entregas posibles a realizar por un camión por lo que el número de etapas de agregación, N , se determina de la siguiente manera

$$N \simeq \log_2 t \quad (4.4)$$

ya que 2^N es el mayor número de puntos agregados en la etapa N (la final).

Es decir, como se observa en la imagen anterior, siempre hay una distancia constante (P0P1, P0P2, P0P3 y P0P4) y una distancia variable (P1P2, P3P4 en la primera, P1P4 y P2P3 en la segunda y P1P2, P0P3 Y P0P4 en la última)

En cada etapa intermedia se determinan los emparejamientos óptimos correspondientes a las distancias mínimas entre los puntos o agregaciones. En la etapa final, los emparejamientos óptimos se determinan de modo que la suma de las longitudes sea mínima. A continuación mostraremos los detalles de este procedimiento con un ejemplo, siendo la formulación del problema de reparto de gasolina la siguiente:

1. Dado un conjunto de n puntos de estación $P_i (i = 1, \dots, n)$ a los que se realizan las entregas desde el punto P_0 (depósito).
2. Se da una matriz de distancias $[c_{i,j}]$ que especifica la distancia entre cada par de puntos (i, j) con $c_{i,j} = c_{j,i}$.
3. Se da un vector de entrega $(D) = (d_i)$ que especifica la cantidad de gasolina que se entrega a cada punto.
4. La capacidad del camión es C donde $C > \max d_i$.
5. Si $x_{i,j} = x_{j,i} = 1$ significa que los puntos i y j están emparejados, y si $x_{i,j} = x_{j,i} = 0$ no lo están, de donde se obtiene la siguiente condición

$$\sum_{j=0, j \neq i}^n x_{i,j} = 1 \quad (i = 1, 2, \dots, n) \quad (4.5)$$

ya que cada punto P_i está conectado con P_0 , o como máximo con otro punto P_j . Además por definición $x_{i,i} = 0 \quad \forall i = 1, 2, \dots, n$.

6. El problema es encontrar los valores de $x_{i,j}$ que hacen la distancia variable total

$$\sum_{i,j=0, i \neq j}^n c_{i,j} x_{i,j} \quad (4.6)$$

mínima en las condiciones especificadas de 2 a 5.

4.1.3. Procedimiento Computacional

Observaciones generales

De acuerdo con la condición 5, $x_{i,j}$ solo puede valer 0 ó 1. Pero en el momento en el que se escribió el artículo, no existía ningún algoritmo apto para resolver problemas de programación entera, aunque los autores comentan que en ese momento R. Gomory estaba desarrollando un algoritmo de esta índole, pero se encontraba en una etapa muy temprana de desarrollo para la evaluación del problema en cuestión. Por lo que los autores deciden relajarlo a un problema de programación lineal, que es más sencillo que uno de programación entera, usando un método muy parecido al que emplearon para resolver el TSP, que está relacionado con el algoritmo para resolver el problema de transporte. Por lo que admitiendo la siguiente condición uno podría determinar “mejores soluciones”

$$0 \leq x_{i,j} \leq 1 \quad (4.7)$$

y luego aplica los métodos de programación lineal conocidos [10]. La posible aparición de valores fraccionarios en la solución indica la existencia de pares o grupos de puntos alternativos. Sin embargo, comentan que la experiencia demostró que el número de estos emparejamientos es pequeño, por lo que el emparejamiento que produce el menor kilometraje se puede encontrar fácilmente por ensayo y error. La solución obtenida después satisface el requisito de que $x_{i,j}$ sea 0 ó 1; sin embargo, el método, no garantiza que se obtenga la solución óptima. Aunque los autores afirman que es probable que la mejor solución obtenida usando este método se aproxime a la óptima a medida que aumenta la cantidad de puntos.

Primera etapa de agregación

El procedimiento detallado lo ilustraremos mejor con el ejemplo numérico que se muestra en la Tabla1. Las entregas se realizan desde el punto P_0 a los puntos P_1, \dots, P_{12} . La entrada en la esquina inferior derecha de cada casilla es la distancia más corta entre los puntos P_i y P_j . Estas entradas son los elementos de la matriz de distancias discutidos en la condición 2. El vector de entrega (D) se muestra a la izquierda de la matriz triangular. Si $C = 6000$, el número de etapas es 2. Los $x_{i,j}$ se introducen en la esquina superior izquierda de cada cuadro. Como inicio todos los puntos de entrega P_1, \dots, P_{12} pueden emparejarse con el depósito P_0 para que haya 12 entradas $x_{0,i} = 1$ con $i = 1, \dots, 12$. Estas 12 entradas constituyen el conjunto básico, en el sentido de la programación lineal, al inicio del cálculo. Durante cada iteración siguiente, exactamente un elemento del conjunto básico se elimina y se reemplaza por un nuevo elemento.

El número total de entradas $x_{i,j}$ básicas permanece constante durante la etapa 1. Las esquinas superiores restantes de cada casilla permanecen vacías o se suministran permanentemente con una estrella. Las casillas vacías constituyen el conjunto no básico de entradas que son todas iguales a 0. Sin embargo, los ceros no se ponen en las casillas para distinguirlos de los ceros que pertenecen al conjunto básico. Una entrada con estrella indica que el emparejamiento de puntos respectivos no es admisible durante las agregaciones de la primera etapa, ya que la demanda combinada de dichos puntos excede la mitad de la capacidad del camión. Como ocurre por ejemplo con la demanda combinada de P_5 y P_8 , $1700 + 1900 = 3600$, que es mayor que $C/2 = 3000$.

Correcciones rápidas

Los autores creen que la solución inicial, en la que cada punto está emparejado con el depósito puede mejorarse fácilmente mediante una serie de correcciones rápidas. Esto es importante ya que el número de iteraciones consecutivas disminuye a medida que disminuye la diferencia entre la solución inicial y la óptima. Por lo que mediante ensayo y error observan que se pueden hacer correcciones rápidas al introducir en la solución entradas no básicas que corresponden a valores de $c_{i,j}$ relativamente pequeños. Por lo tanto, $c_{6,7} = 7$ es relativamente pequeño y puede introducirse en el conjunto básico ingresando el valor θ , aún no determinado, en la esquina superior izquierda correspondiente. Para satisfacer las ecuaciones (4.5), las sumas de las entradas básicas para cualquier $i = 1, 2, \dots, n$ debe ser igual a 1. Dado que la matriz de distancias es simétrica, se deduce que la suma de los valores básicos en la fila y columna 6 en la matriz triangular que se muestra en la Tabla1 también debe ser igual a 1. Lo mismo ocurre con la fila y la columna 7. Por lo tanto se hacen las siguientes entradas:

$$x_{6,7} = \theta \quad \text{entrada no básica } x_{6,7} = 0 \text{ incrementada}$$

Como consecuencia a estas correcciones, la distancia total se ha reducido de 364 a 170 unidades.

Cabe señalar que se ha obtenido una mejora notable, pero que no es un método fiable, ya que podría haber otros cambios de base que mejoraran más rápido la función objetivo.

Costos marginales

Cuando un número suficiente de pares de puntos con una distancia pequeña entre ellos entra en la solución, será cada vez más complicado que pares de puntos entren en la solución sin calcular la distancia total en cada caso.

Por lo tanto se necesita un criterio que permita aceptar o rechazar una entrada básica.

Este criterio lo proporciona la función delta definida como:

$$\delta_{i,j}^{(n)} = \pi_i^{(n)} + \pi_j^{(n)} - c_{i,j} \quad (4.8)$$

donde $\pi_i^{(n)}$ y $\pi_j^{(n)}$ son constantes determinadas por la n -ésima iteración. Por definición $\pi_i^{(n)}$ y $\pi_j^{(n)}$ son determinadas de la siguiente manera:

$$\delta_{i,j}^{(n)} = 0 \quad (4.9)$$

$\forall c_{i,j}$ correspondientes a las entradas básicas. Para las entradas no básicas:

$$\delta_{i,j}^{(n)} \leq 0 \quad (4.10)$$

Encontramos una gran similitud con los costos marginales que se dan en el problema de transporte estudiados en la asignatura de investigación operativa, por lo que se tratarán de la misma manera. Veremos en un apartado posterior, que efectivamente, son los costos marginales.

La función delta indica cuánto disminuirá la distancia total por incremento unitario de una entrada no básica $x_{i,j}$, si $\delta_{i,j}^{(n)} < 0$ la distancia total aumenta si se aumenta el valor no básico correspondiente.

Por lo tanto al escanear la matriz de la Tabla2, los cuadros con $\delta_{i,j}^{(n)} < 0$ no se tienen en cuenta a menos que $\delta_{i,j}^{(n)}$ sea negativo para todas las entradas no básicas. En este caso todas las posibles entradas no básicas darían una solución que aumentaría la distancia total representada por el conjunto básico de entradas. Llegados a este punto, el conjunto obtenido representaría la mejor solución.

Sin embargo, siempre que haya al menos un $\delta_{i,j}^{(n)} > 0$ es posible una reducción de la distancia total. Si hay más de un $\delta_{i,j}^{(n)} > 0$ la mayor reducción de la distancia total posible relativa a cada unidad de $x_{i,j}$ se consigue con el mayor $\delta_{i,j}^{(n)}$, ya que como se trata de un problema de mínimo, estaríamos hablando no de los costos, sino de los -costos

Se elige como entrada básica aquella que tenga mayor $\delta_{i,j}^{(n)}$. En caso de empate, se elegirá la primera de ellas.

Las constantes $\pi_i^{(n)}$ y $\pi_j^{(n)}$ en la función delta (4.8) pueden ser determinadas de la condición (4.9). Esto da lugar a la siguiente ecuación:

$$\pi_i^{(n)} + \pi_j^{(n)} - c_{i,j} = 0. \quad (4.11)$$

En la que uno puede establecer arbitrariamente $\pi_0 = 0$. Mientras que los autores afirman que cualquier elección de un entero ≥ 0 es admisible, demostraremos más adelante que no lo es. Por ello, tomaremos $\pi_0 = 0$.

Como hay 12 entradas en el ejemplo de la Tabla2, hay 12 ecuaciones de tipo (4.11) a partir de las cuales se pueden determinar los 12 $\pi_i^{(n)}$ -valores correspondientes a cada punto de entrega P_i . Estos 12 valores y $\pi_0 = 0$ se ingresan en la esquina inferior derecha de los cuadros que tienen las identificaciones de puntos como se muestra en la Tabla2.

$$\begin{aligned} \pi_0 &= 0 \\ \pi_0 + \pi_1 - c_{0,1} &= 0 + \pi_1 - 9 = 0 \implies \pi_1 = 9 \\ \pi_1 + \pi_2 - c_{1,2} &= 9 + \pi_2 - 5 = 0 \implies \pi_2 = -4 \\ \pi_0 + \pi_3 - c_{0,3} &= 0 + \pi_3 - 21 = 0 \implies \pi_3 = 21 \\ \pi_3 + \pi_4 - c_{3,4} &= 21 + \pi_4 - 10 = 0 \implies \pi_4 = -11 \\ \pi_0 + \pi_5 - c_{0,5} &= 0 + \pi_5 - 22 = 0 \implies \pi_5 = 22 \\ \pi_0 + \pi_6 - c_{0,6} &= 0 + \pi_6 - 25 = 0 \implies \pi_6 = 25 \\ \pi_6 + \pi_7 - c_{6,7} &= 25 + \pi_7 - 7 = 0 \implies \pi_7 = -18 \\ \pi_0 + \pi_8 - c_{0,8} &= 0 + \pi_8 - 36 = 0 \implies \pi_8 = 36 \\ \pi_0 + \pi_9 - c_{0,9} &= 0 + \pi_9 - 38 = 0 \implies \pi_9 = 38 \\ \pi_0 + \pi_{10} - c_{0,10} &= 0 + \pi_{10} - 42 = 0 \implies \pi_{10} = 42 \\ \pi_0 + \pi_{11} - c_{0,11} &= 0 + \pi_{11} - 50 = 0 \implies \pi_{11} = 50 \\ \pi_{11} + \pi_{12} - c_{11,12} &= 50 + \pi_{12} - 10 = 0 \implies \pi_{12} = -40 \end{aligned}$$

Después de obtener los valores de $\pi_i^{(n)}$, los $\delta_{i,j}^{(n)}$ se calculan a partir de (4.8) y $\max \delta_{i,j}^{(n)} = \delta_{r,s}^{(n)}$ determinado de la forma descrita anteriormente.

$$\begin{aligned} \delta_{0,1} &= 0 + 9 - 9 = 0 \\ \delta_{0,2} &= 0 - 4 - 14 = -18 \\ \delta_{1,2} &= 9 - 4 - 5 = 0 \\ \delta_{0,3} &= 0 + 21 - 21 = 0 \\ \delta_{1,3} &= 9 + 21 - 12 = 18 \\ \delta_{0,4} &= 0 - 11 - 23 = -34 \\ \delta_{1,4} &= 9 - 11 - 22 = -24 \\ \delta_{3,4} &= 21 - 11 - 10 = 0 \\ \delta_{0,5} &= 0 + 22 - 22 = 0 \\ \delta_{1,5} &= 9 + 22 - 21 = 10 \end{aligned}$$

$$\delta_{0,6} = 0 + 25 - 25 = 0$$

$$\delta_{1,6} = 9 + 25 - 24 = 10$$

$$\delta_{3,6} = 21 + 25 - 30 = 16$$

$$\delta_{4,6} = -11 + 25 - 28 = -14$$

$$\delta_{0,7} = 0 - 18 - 32 = -50$$

$$\delta_{1,7} = 9 - 18 - 31 = -40$$

$$\delta_{2,7} = -4 - 18 - 26 = -48$$

$$\delta_{3,7} = 21 - 18 - 27 = -24$$

$$\delta_{4,7} = -11 - 18 - 25 = -54$$

$$\delta_{5,7} = 22 - 18 - 10 = -6$$

$$\delta_{6,7} = 25 - 18 - 7 = 0$$

$$\delta_{0,8} = 0 + 36 - 36 = 0$$

$$\delta_{0,9} = 0 + 38 - 38 = 0$$

$$\delta_{1,9} = 9 + 38 - 37 = 10$$

$$\delta_{7,9} = -18 + 38 - 16 = 4$$

$$\delta_{0,10} = 0 + 42 - 42 = 0$$

$$\delta_{1,10} = 9 + 42 - 41 = 10$$

$$\delta_{4,10} = -11 + 42 - 29 = 2$$

$$\delta_{6,10} = 25 + 42 - 17 = 50$$

$$\delta_{7,10} = -18 + 42 - 10 = 14$$

$$\delta_{0,11} = 0 + 50 - 50 = 0$$

$$\delta_{1,11} = 9 + 50 - 49 = 10$$

$$\delta_{7,11} = -18 + 50 - 18 = 14$$

$$\delta_{0,12} = 0 - 40 - 52 = -92$$

$$\delta_{1,12} = 9 - 40 - 51 = -82$$

$$\delta_{2,12} = -4 - 40 - 46 = -90$$

$$\delta_{3,12} = 21 - 40 - 39 = -58$$

$$\delta_{4,12} = -11 - 40 - 29 = -80$$

$$\delta_{5,12} = 22 - 40 - 30 = -48$$

$$\delta_{6,12} = 25 - 40 - 27 = -42$$

$$\delta_{7,12} = -18 - 40 - 20 = -78$$

$$\delta_{8,12} = 36 - 40 - 16 = -60$$

$$\delta_{9,12} = 38 - 40 - 20 = -22$$

$$\delta_{10,12} = 42 - 40 - 10 = -8$$

$$\delta_{11,12} = 50 - 40 - 10 = 0$$

La entrada no básica $x_{r,s}^{(n)}$ se establece igual a θ y las entradas básicas se corrigen cuando sea necesario restando θ de tal manera que la suma de las entradas en la fila y la columna r , así como en la fila y la columna s , sean por separados iguales a 1.

En la Tabla2 el $\max \delta_{i,j} = \delta_{6,10} = 25 + 42 - 17 = 50$, en consecuencia, la entrada no básica $x_{6,10} = 0$ es reemplazada por $x_{6,10} = \theta$. Las correcciones θ de las entradas básicas se hacen como se muestra. Cabe señalar que las correcciones θ no se pueden aplicar a la entrada básica $x_{6,7} = 1$ ya que no hay otra entrada básica en la séptima fila y la séptima columna a la que aplicarle la corrección correspondiente. A partir de este punto, el procedimiento es el mismo que se ha explicado anteriormente, excepto por una condición; se determina el máximo valor de θ que deja las entradas básicas no negativas y de aquellas entradas básicas que son 0, pero que se hubieran vuelto negativas si se hubiera elegido θ más grande, se elimina del conjunto básico la entrada que corresponde al mayor $c_{i,j}$. En este caso el $c_{i,j}$ correspondiente a $x_{0,11} = 0$ es mayor que el $c_{i,j}$ asociado a $x_{0,6}$, el valor básico $x_{0,6}$ debe eliminarse en lugar de $x_{0,11}$ ya que un pequeño valor de θ positivo habría impulsado a que $x_{0,6}$ fuese negativo, pero no hubiese afectado a $x_{0,11}$.

Si un mayor valor de θ hubiera hecho que $x_{0,6}$ y $x_{0,11}$ fuesen negativos, entonces $x_{0,11}$ debería eliminarse en lugar de $x_{0,6}$. Esta disposición a la eliminación de entradas básicas que son 0 es necesaria para evitar ciclos.

Al iterar este proceso, las entradas básicas se incorporan al conjunto básico hasta que no sea posible una mejora adicional. Como se ha señalado anteriormente, este es el caso cuando la función delta es negativa o cero para todas las entradas básicas. Esto concluye las agregaciones de la primera etapa, en las cuales se emparejaron los puntos cuya demanda combinada no excede la mitad de la carga del camión. Estos emparejamientos se muestran en la Figura1; en la cual observamos 2 agregaciones que contienen un punto de estación y el depósito, y otras 5 con dos puntos de estación.

Segunda etapa de agregación

Los pares de puntos obtenidos en la agregación de la primera etapa para el ejemplo, ahora se pueden combinar entre sí, sin exceder la capacidad del camión. Por lo tanto, en la agregación de la segunda etapa no habrá, en general, entradas con estrella. El problema es combinar las agregaciones de la primera etapa de tal manera que la distancia total se convierta en un mínimo. El primer paso será configurar una tabla de distancias triangular para pares de agregados similar a la Tabla1.

Denominando a las agregaciones de primera etapa como A_1, \dots, A_7 y el punto P_0 con A_0 , uno obtiene la matriz triangular de la Tabla3.

Las distancias de la esquina inferior derecha de la columna A_0 corresponden a las rutas mínimas desde el depósito a cada punto en el agregado respectivo y de regreso al depósito. Las entradas restantes representan las distancias de las rutas mínimas desde el depósito a cada punto de pares de agregados, A_i y A_j , y de regreso al terminal.

Estas rutas de bucle cerrado se pueden determinar fácilmente a partir de la matriz de distancias, si se tiene cuidado de que ningún bucle se cruce por sí mismo (esto solo es posible analizarlo si al inicio de cada problema tenemos el grafo con la situación de cada punto), que en realidad se trata de la resolución de un TSP, que emplea un método

que dará lugar a una solución heurística. Por ejemplo la ruta mínima que involucra los agregados A_4 y A_5 es $P_0P_6P_7P_{10}P_9P_0$ y no $P_0P_6P_{10}P_7P_9P_0$.

El procedimiento para encontrar la combinación de agregados que produce el kilometraje mínimo es exactamente el mismo que el utilizado para las agregaciones de la primera etapa. La Tabla3 muestra la solución óptima obtenida. Si no fuera por la aparición de valores fraccionarios para algunos $x_{i,j}$, se resolvería el problema de la segunda etapa.

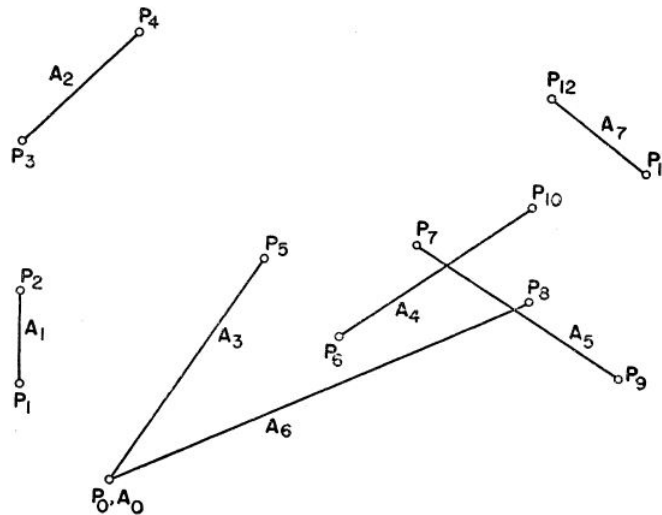


Figura 1

Tratamiento de los valores fraccionarios de $x_{i,j}$

Según las relaciones (4.7) se permiten los valores fraccionarios durante el cálculo y no se requiere un tratamiento especial a menos que aparezcan en la solución óptima.

Un valor fraccionario para cualquier $x_{i,j}$ en la solución óptima significa que no se ha llegado a una decisión sobre si los puntos P_i y P_j o los agregados A_i y A_j se deben unir o no.

Las soluciones fraccionarias siempre implican uno o más grupos de $x_{i,j}$ fraccionarios, cada uno de ellos compuesto por un número impar de entradas básicas.

A_0									
0									
28	A_1								
	17								
54	$1/2$	A_2							
	54	37							
44	$1/2$	$1/2$	A_3						
	52	72	35						
84				A_4					
	92	102	84	45					
86				0	A_5				
	94	110	87	92	47				
72			0	1		A_6			
	80	102	74	84	86	39			
112					1		A_7		
	120	120	112	112	112	112	65		

4.2. Consideraciones acerca del artículo

- La experiencia con este método ha demostrado que se pueden obtener resultados similares en otros casos numéricos, particularmente si la demanda de la estación no difiere demasiado; es por ello que se impone la condición de que en cada etapa no se exceda cierta capacidad, es decir, siendo N el número de etapas necesario para resolver cada problema tenemos lo siguiente:

Etapla 1: se resuelve el problema en el que cada camión visita 1 ó 2 nodos y cuyo reparto no exceda la capacidad $\frac{C2}{2^N}$

Etapla 2: se resuelve el problema en el que cada camión visita 1 ó 2 agregaciones (colecciones de 1 ó 2 nodos que habían sido óptimos en la etapa 1), y cuyo reparto no exceda la capacidad $\frac{C2^2}{2^N}$

.

.

.

Etapla N : se resuelve el problema en el que cada camión visita 1 ó 2 agregaciones que han sido óptimas en la etapa $N - 1$ y cuyo reparto no exceda la capacidad $\frac{C2^N}{2^N} = C$

Ya que como se ha dicho anteriormente $2^N \geq t$, y un camión que visite $t + 1$ nodos excede la capacidad.

Por ello, al principio de la primera etapa se destacan aquellas entradas cuya demanda combinada no exceda la capacidad correspondiente, ya que esto garantiza que al final de la etapa N no se haya superado la capacidad C .

Estas condiciones solo tiene sentido imponerlas cuando las demandas de cada estación sean similares, ya que si estas difiriesen mucho, sus demandas combinadas podrían equilibrarse y terminar sin superar la capacidad C .

No obstante, si se quitase la restricción de que en cada etapa de agregación sólo los clientes cuya carga combinada no excedan la carga total asignada, el problema sería el mismo, a diferencia de que en cada etapa se permitiría unir dos puntos cualesquiera cuya carga combinada no exceda C . Dicho algoritmo fue planteado por Clarke y Wright en 1964 [11], con el cual se comprobó que en los casos en los cuales la carga difiere mucho, dicho método proporcionaba mejores resultados que el método de Ramser y Dantzig, pero si la carga es similar es mejor usar el algoritmo que plantearon Ramser y Dantzig, ya que para la resolución del problema planteado anteriormente se obtiene una solución de 312 unidades, mientras que con el algoritmo de Dantzig y Ramser se obtiene una solución de 294 unidades, una solución más próxima a la verdadera óptima.

También se conjetura que la diferencia entre la distancia de la mejor solución obtenida y de la solución óptima disminuye a medida que aumenta el número de puntos de estación.

- Una vez visto el trato que se le da a estos costos, podemos afirmar que efectivamente para el caso de $\pi_0 = 0$ se trata de los costos marginales, ya que dado un problema P, en el cual $x_{i,j} = x_{j,i}$, tomaremos, por tanto, $j < i$, para este caso en particular.

$$\text{Min } z = \sum_{0 \leq j < i \leq n} c_{i,j} x_{i,j} \quad (4.12)$$

sujeto a

$$\begin{aligned} \sum_{0 \leq j < i} x_{i,j} + \sum_{n \geq j > i} x_{i,j} &= 1 \quad \forall i = 1, \dots, n \\ x_{i,j} &\geq 0 \end{aligned} \quad (4.13)$$

Las ecuaciones 4.12 y 4.13 se corresponden con las ecuaciones 4.6 y 4.5 respectivamente. Encontramos otro problema, P' , que en este dominio es equivalente, dicho problema se formula de la siguiente manera

$$\text{Min } z = \sum_{0 \leq j < i \leq n} c_{i,j} x_{i,j} + \sum_{i=1}^n \pi_i \left(1 - \sum_{0 \leq j < i} x_{i,j} - \sum_{i < j \leq n} x_{i,j} \right)$$

sujeto a

$$\begin{aligned} \sum_{0 \leq j < i} x_{i,j} + \sum_{n \geq j > i} x_{i,j} &= 1 \quad \forall i = 1, \dots, n \\ x_{i,j} &\geq 0 \end{aligned}$$

que despejando dará lugar a

$$\text{Min } z = \sum_{0 \leq j < i \leq n} x_{i,j} (c_{i,j} - \pi_i - \pi_j) + \sum_{i=1}^n \pi_i \Rightarrow \text{Min } z = \sum_{0 \leq j < i \leq n} x_{i,j} (-\delta_{i,j}) + \sum_{i=1}^n \pi_i$$

donde $\delta_{i,j} = 0$ para las variables básicas, por lo tanto el primer sumatorio solo depende de las variables no básicas, pero dichas variables tienen un valor de $x_{i,j} = 0$ por lo que el valor de z en la solución básica es $\sum_{i=1}^n \pi_i$.

Mientras que para los casos en los que $\pi_0 \neq 0$ y positivo no se puede afirmar que se trate de los costos marginales, como se hace en el artículo de Dantzig y Ramser, ya que se puede comprobar que para valores de $\pi_0 \neq 0$ dichos costos varían (en el caso de las variables no básicas), incluso llegando a cambiar de signo, como podemos comprobar tomando por ejemplo $\pi_0 = 20$ para el caso resuelto en páginas anteriores, con el cual se tienen los siguientes resultados:

$$\pi_2 = 16 \Rightarrow \delta_{0,2} = 20 + 16 - 14 = 22,$$

$$\pi_1 = -11, \pi_3 = 1 \Rightarrow \delta_{1,3} = -11 + 1 - 12 = -22.$$

Capítulo 5

Un algoritmo de ramificación y acotación.

El primer algoritmo de ramificación y acotación para el TSP fue formulado por Little, Murty, Sweeney y Karel en 1963 [12], en el cual se basaron Christofides y Eilon [13] para plantear el primero orientado a la resolución de CVRPs en 1969.

Sin embargo, nosotros analizaremos el descrito por Laporte, Mercure y Nobert en 1986, por ser más sofisticado. Su objetivo fue desarrollar un algoritmo exacto para la resolución del problema de rutas de vehículos con capacidad asimétrico. El problema se resuelve mediante una rama y un árbol, de los cuales van derivando subproblemas, problemas de asignación sujetos a algunas restricciones.

Con este algoritmo los autores resolvieron de forma óptima CVRPs asimétricos que involucran hasta 260 nodos.

5.1. Descripción

Se considera el grafo $G = (V, A)$ donde $V = \{0, \dots, n\}$ es el conjunto de nodos, con el depósito en el nodo 0 y A el conjunto de arcos dirigidos entre nodos. El problema consiste en establecer rutas para vehículos idénticos, cada uno con capacidad C y costo fijo f de tal manera que

- Cada vehículo comienza y termina en el depósito.
- Cada ciudad, excepto el depósito, es visitada exactamente una vez por un vehículo.
- La suma de las demandas en una ruta no exceda la capacidad del vehículo C
- Se minimiza la suma de los costos fijos y enrutamiento.

5.2. Formulación

En el siguiente algoritmo [3] veremos la relación entre el VRP y el m-TSP, con el depósito en el vértice 0 y m vehículos disponibles en el depósito. El m-TSP consiste en establecer m rutas con el menor costo posible y que terminen en el depósito, de forma que cada vértice sea visitado una única vez.

Dado un límite superior m_U de m , el m-TSP se puede transformar en un 1-TSP de la siguiente manera:

- Aumentando el número de nodos introduciendo $m_U - 1$ depósitos artificiales, donde $n' = n + m_U - 1$, $V' = \{0, 1, 2, \dots, n'\}$, $V^* = (V' \setminus V) \cup \{0\}$ y $A' = A \cup \{(i, j) : i, j \in V', i \neq j, i \text{ or } j \in V' \setminus V\}$.
- La matriz de costos asociada a A' está definida como:

$$c'_{ij} = \begin{cases} c_{ij} & (i, j \in V) \\ c_{i0} & (i \in V \setminus \{0\}, j \in V' \setminus V) \\ c_{0j} & (i \in V' \setminus V, j \in V \setminus \{0\}) \\ \gamma & (i, j \in V^*) \end{cases} \quad (5.1)$$

donde el valor de γ depende de la variante del problema, tomándolo de tal forma que:

$\gamma = \infty$ para calcular la mínima distancia para m_U vehículos.

$\gamma = 0$ para calcular la distancia mínima de a los sumo m_U vehículos.

$\gamma = -\infty$ para calcular la distancia mínima para el número mínimo de vehículos.

Sea x_{ij} con $i \neq j$ una variable binaria que tome el valor 1 cuando el arco (i, j) se encuentre en la solución y 0 en caso contrario. Si $d_i + d_j \geq C$, i y j no pueden pertenecer a la misma ruta y no es necesario definir x_{ij} .

$$\text{Minimizar } \sum_{i \neq j} c'_{ij} x_{ij} \quad (5.2)$$

sujeto a:

$$\sum_{i=1}^{n'} x_{ij} = 1 \quad (j = 1, 2, \dots, n'), \quad (5.3)$$

$$\sum_{j=1}^{n'} x_{ij} = 1 \quad (i = 1, 2, \dots, n'), \quad (5.4)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - \gamma(S) \quad (S \subset V \setminus \{0\}; |S| \geq 2), \quad (5.5)$$

$$x_{ij} \in [0, 1] \quad (i, j = 1, 2, \dots, n'; i \neq j). \quad (5.6)$$

Con esta definición dejamos la diagonal vacía y las restricciones (5.5) corresponden a la eliminación de subciclos.

Estas restricciones se obtienen observando que para cualquier $S \subset V \setminus \{0\}, |S| \geq 2$

$$\sum_{i \in V' \setminus S} \sum_{j \in S} x_{ij} \geq \gamma(S)$$

donde

$$\gamma(S) = \left\lceil \frac{\sum_{i \in S} d_i}{C} \right\rceil. \quad (5.7)$$

Una vez resuelto el problema debemos deshacer los cambios y considerar un arco (i, j) perteneciente a la solución como:

- Si $i \in V \setminus \{0\}, j \in V' \setminus V$ reemplazamos (i, j) por $(i, 0)$.
- Si $i \in V' \setminus V, j \in V \setminus \{0\}$ reemplazamos (i, j) por $(0, j)$.
- Si $i, j \in V' \setminus V$ eliminamos el arco (i, j) .

5.3. Algoritmo

A continuación, lo que se hace es resolver el problema con las restricciones (5.3) y (5.4) y luego manejar las restricciones de eliminación de subtours por medio de un árbol, fijando las variables $x_{i,j}$ como 0 ó 1.

Como todos los subproblemas son problemas de asignación modificados darán una solución entera. Es preciso mencionar, que posteriormente se analizará el algoritmo con un ejemplo.

Definimos la siguiente notación:

z^* : costo de la mejor solución posible identificada hasta ahora.

z_h : el valor de la función objetivo del problema de asignación modificado en h del árbol de búsqueda.

z'_h : un límite inferior de z_h .

I_h : el conjunto de arcos incluidos en el nodo h del árbol de búsqueda

E_h : el conjunto de arcos excluidos en el nodo h del árbol de búsqueda.

Empleando la siguiente notación, el algoritmo se puede definir mediante los siguientes pasos:

Paso 0 (Inicialización):

Se obtiene un primer valor para z^* mediante un método heurístico apropiado, los autores emplean el método de Clarke y Wright. Se crea el primer nodo del árbol de búsqueda y se establece $I_1 = E_1 = \emptyset$ y se obtiene z_1 resolviendo el problema de asignación en cuestión. Si $z_1 \geq z^*$ se termina. Si la solución no contiene subciclos, hemos llegado a la solución óptima y se termina. De lo contrario se inserta el nodo 1 en la cola y se va al paso 1.

Paso 1 (Selección del nodo):

Si la cola está vacía se finaliza. En caso contrario, se selecciona el siguiente nodo (h) de la cola. En la primera versión de profundidad del algoritmo, el nodo h se selecciona de acuerdo a la regla LIFO, el último en entrar es el primero en salir, esta regla reduce la memoria; mientras que en la primera versión de amplitud, la ramificación se hace siempre en el nodo que tiene z_h más bajo; que suele minimizar el número de subproblemas resueltos. Vaya al paso 2.

Paso 2 (Ramificación):

La solución encontrada en el nodo h no cumple todas las restricciones y debe eliminarse mediante la partición del subproblema actual en subproblemas descendientes j caracterizados por el conjunto de arcos incluidos y excluidos, I_j y E_j respectivamente. La ramificación se hace al romper un subtour “ilegal” de acuerdo con una regla de partición. Las reglas para la elección del subtour y su partición se describirán más adelante.

Para cada subproblema j se ejecutan del paso 3 al 6. Luego se ejecuta el paso 7.

Paso 3 (Acotación):

Se calcula una cota inferior z'_j de z_j . Si $z'_j < z^*$, se continua con el paso 4. De lo contrario, se considera el siguiente j y se repite el paso 3, puesto que las soluciones obtenidas al ramificar dicho nodo empeorarán la función objetivo.

Paso 4 (Solución del problema de asignación):

Se resuelve el subproblema de asignación asociado al nodo j , un problema de asignación restringido por los conjuntos I_j y E_j . Si $z_j \geq z^*$ se considera el siguiente j y se continua con el paso 3. En caso contrario se va al paso 5.

Paso 5 (Verificación de factibilidad):

Se verifica si la solución actual contiene subtours ilegales, cualquier subtour desconectado de V^* es ilegal, además cualquier cadena entre dos depósitos con una carga total que exceda C corresponde a un subtour ilegal de G . Si la solución no contiene subtours ilegales, se modifica $z^* = z_j$ y se guarda el recorrido; si $z^* = z_h$ se va al paso 7; se considera el próximo j y se va al paso 3. Si la solución contiene al menos un subtour ilegal se va al paso 6.

Paso 6 (Verificación de la inclusión del nodo j en la cola):

La solución actual contiene subtours con nodos que pertenecen a los subconjuntos S_k que son un subconjunto de $V \setminus \{0\}$. Para cada k es válido excluir hasta $\gamma(S_k)$ arcos del subtour. Si al menos un subtour secundario es tal que $\gamma(S_k)$ excede el número de sus arcos que aún no están incluidos en I_j , cualquier ramificación del subproblema j no tendrá solución factible. Mediante este paso se pretende hacer referencia a las rupturas de subtours (3.13). Por lo tanto, el nodo j se inserta en la cola solo si

$$\gamma(S_k) \leq |S_k| - |(S_k \times S_k) \cap I_j| \quad \forall k.$$

Paso 7 (Recuperación de memoria):

Si los requisitos de memoria estimados para los siguientes subproblemas exceden la memoria disponible, se eliminan de la cola todos los nodos h para los que $z_h \geq z^*$. Se vuelve al paso 1.

5.4. Regla de partición

Considere un subtour ilegal que se ha identificado en el paso 2 del algoritmo. Supongamos que ese subtour contiene un conjunto T de arcos que no pertenecen a I_h

$$T = \{(\alpha_1, \beta_1), \dots, (\alpha_M, \beta_M)\}.$$

Estos arcos se enumeran en el orden en el que aparecen en el subtour, donde (α_1, β_1) es el primer arco en orden lexicográfico.

Regla : A los M nodos descendientes (j) del nodo h , se asocian los siguientes conjuntos de arcos incluidos y excluidos

$$I_j = \begin{cases} I_h & (j = 1) \\ I_h \cup \{(\alpha_u, \beta_u) : u = 1, \dots, j-1\} & (j = 2, \dots, M) \end{cases} \quad (5.8)$$

$$E_j^0 = E_h \cup \{(\alpha_j, \beta_j)\} \quad (j = 1, \dots, M) \quad (5.9)$$

Cuando se usa esta regla, la bifurcación se lleva a cabo desde el subtour que contiene el menor valor de M , es decir, el menor número de descendientes inmediatos. En el CVRP, es válido adjuntar a E_j^0 los siguientes conjuntos $H(E_j^0)$ y $W(I_j)$ de arcos excluidos.

1. Réplicas de arcos excluidos entre V^* y $V' \setminus V^*$

$$\begin{aligned} H(E_j^0) = & \cup_{k \in V^*} [\{(\alpha_i, k) : (\alpha_i, \beta_i) \in E_j^0, \alpha_i \in V' \setminus V^*, \beta_i \in V^*\} \\ & \cup \{(k, \beta_i) : (\alpha_i, \beta_i) \in E_j^0, \alpha_i \in V^*, \beta_i \in V' \setminus V^*\}] \end{aligned}$$

2. Los arcos se excluyen para evitar que se produzcan subtours que tengan una carga total superior a C : En el subproblema j , V' se puede dividir en $Q(V') = \{S_1, \dots, S_t\}$ donde S_1, \dots, S_t son subconjuntos de nodos enlazados por los arcos de I_j . A cada subconjunto S_t le corresponde una cadena con un peso

$$w(S_t) = \sum_{i \in S_t} d_i$$

y dos nodos finales p_t y q_t que coinciden si $|S_t| = 1$.

Sea

$$e(S_t) = \{p_t, q_t\}.$$

Cualquiera de los subconjuntos de nodos S_t y S_v no pueden pertenecer a la misma ruta en la solución óptima si su carga total excede C . Por lo tanto, es válido excluir los arcos que unen los nodos finales de las cadenas asociadas con S_t y S_v . Esto se hace estableciendo

$$W(I_j) = \{(i, j) \in [V \setminus \{0\}]^2 : i \in e(S_t), j \in e(S_v), S_t, S_v \in Q(V'), w(S_t) + w(S_v) > C\}.$$

$W(I_j)$ corresponde al conjunto de arcos forzados a 0. Vale la pena notar que el número de arcos incluidos en $W(I_j)$ está relacionado con el número de cadenas en I_j y su peso. Dado que los arcos se incluyen sucesivamente en el I_j s, tal y como aparecen en el subtour, es muy probable que se creen cadenas de arcos de esta manera.

3. E_j^0 se aumenta después en E_j mediante la inclusión de todos los arcos que sin duda crearían subtours ilegales si aparecieran en la solución.

5.5. Ejemplo

Veamos la aplicación de este algoritmo al problema de Dantzig y Ramser planteado en el capítulo 4. En esta sección se aplicará el algoritmo paso a paso para los siete primeros problemas (versión en amplitud), para ver el funcionamiento del algoritmo. Una vez visto el funcionamiento del algoritmo se analizarán algunos subproblemas concretos en los cuales se ven pasos del algoritmo no vistos anteriormente. Sin embargo, en el Anexo veremos la aplicación del algoritmo completa a los siete problemas comentados anteriormente. A partir de este punto, como no es posible hacer a mano una ejecución completa del algoritmo, y teniendo en cuenta que ya conocemos cual va a ser la solución óptima (citada en el artículo de Dantzig y Ramser analizado anteriormente), se irá seleccionando el nodo más conveniente para llegar antes a la solución óptima (profundidad).

Por otra parte, en la práctica no se calculará una cota inferior de la solución (Paso 3). El motivo es que el paso 3 solo juega un papel relevante si la cota inferior obtenida es superior a z^* . En este caso nos ahorraría ejecutar el paso 4, que es más costoso. En caso contrario el paso 3 es redundante con el paso 4. Como en nuestro ejemplo, debido a que no se solucionará el problema al completo, solo se supera z^* a lo sumo una única vez, nos compensa no ejecutar el paso 3 para ahorrar tiempo.

Para comenzar a resolver el problema debemos añadir tres depósitos artificiales ya que $\frac{\sum_{i=1}^{12} d_i}{C} = \frac{91}{30}$, por lo tanto el número mínimo de vehículos necesarios es 4. Denotamos los depósitos como D1, D2, D3 y D4, y los clientes como $P1, \dots, P12$.

Problema 1

Paso 0: Sea $z^* = 294$ obtenido por el algoritmo heurístico de Dantzig y Ramser, que se ha estudiado anteriormente. Tomamos los conjuntos $I_1 = \emptyset = E_1$ y resolvemos el primer problema de asignación para el que hemos usado WinQSB y Lingo, cabe señalar que el ejemplo que estamos resolviendo no es dirigido, de ahí que la matriz sea simétrica, y que las entradas que contienen un 1000 se han tomado de esa manera para penalizar dichos costos y que no se puedan asignar dichos nodos.

From \ To	deposito1	deposito2	deposito3	deposito4	nodo1	nodo2	nodo3	nodo4	nodo5	nodo6	nodo7	nodo8	nodo9	nodo10	nodo11	nodo12
deposito1	1000	1000	1000	1000	9	14	21	23	22	25	32	36	38	42	50	52
deposito2	1000	1000	1000	1000	9	14	21	23	22	25	32	36	38	42	50	52
deposito3	1000	1000	1000	1000	9	14	21	23	22	25	32	36	38	42	50	52
deposito4	1000	1000	1000	1000	9	14	21	23	22	25	32	36	38	42	50	52
nodo1	9	9	9	9	1000	5	12	22	21	24	31	35	37	41	49	51
nodo2	14	14	14	14	5	1000	7	17	16	23	26	30	36	36	44	46
nodo3	21	21	21	21	12	7	1000	10	21	30	27	37	43	31	37	39
nodo4	23	23	23	23	22	17	10	1000	19	28	25	35	41	29	31	29
nodo5	22	22	22	22	21	16	21	19	1000	9	10	16	22	20	28	30
nodo6	25	25	25	25	24	23	30	28	9	1000	7	11	13	17	25	27
nodo7	32	32	32	32	31	26	27	25	10	7	1000	10	16	10	18	20
nodo8	36	36	36	36	35	30	37	35	16	11	10	1000	6	6	14	16
nodo9	38	38	38	38	37	36	43	41	22	13	16	6	1000	12	12	20
nodo10	42	42	42	42	41	36	31	29	20	17	10	6	12	1000	8	10
nodo11	50	50	50	50	49	44	37	31	28	25	18	14	12	8	1000	10
nodo12	52	52	52	52	51	46	39	29	30	27	20	16	20	10	10	1000

Para este problema obtenemos la siguiente solución:

$$D1 \rightarrow P1 \rightarrow D2 \rightarrow P2 \rightarrow D3 \rightarrow P3 \rightarrow P4 \rightarrow D4 \rightarrow P5 \rightarrow D1$$

$$P6 \rightarrow P7 \rightarrow P6$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$$

Con $z_1 = 198$.

La solución obtenida no es factible ya que contiene tres subtours ilegales, porque no pasan por el depósito, $P6 \rightarrow P7 \rightarrow P6$, $P8 \rightarrow P9 \rightarrow P8$ y $P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$. Añadimos el nodo a la cola. Como solo está el nodo 1 en la cola lo seleccionamos (Paso 1). Particionamos por el que tenga menor número de arcos, sin tener en cuenta los incluidos (Paso 2). Es decir, el primero y el segundo tienen dos aristas libres y el tercero tres. Partimos por cualquiera de los dos primeros, elegimos el primero. Examinados los tres puntos de la regla de partición obtenemos dos subproblemas, con sus correspondientes conjuntos de incluidos y excluidos:

- **Problema 1.1** $\rightarrow I_{1.1} = \emptyset$ y $E_{1.1} = \{(P6, P7)\}$
- **Problema 1.2** $\rightarrow I_{1.2} = \{(P6, P7)\}$ y $E_{1.2} = \{(P7, P6)\}$

Para cada subproblema ejecutamos los pasos 4 al 6.

Problema 1.1

Para el Problema de asignación 1.1 tomamos

$$I_{1.1} = \emptyset$$

$$E_{1.1} = \{(P6, P7)\}$$

y lo resolvemos (Paso 4), para el cual obtenemos $z_{1.1} = 200$ y la siguiente asignación

$$D1 \rightarrow P1 \rightarrow D1$$

$$D2 \rightarrow P2 \rightarrow D2$$

$$D3 \rightarrow P3 \rightarrow D3$$

$$D4 \rightarrow P4 \rightarrow D4$$

$$P5 \rightarrow P7 \rightarrow P6 \rightarrow P5$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P12 \rightarrow P11 \rightarrow P10$$

Solución que no es factible porque contiene tres subtours ilegales (Paso 5), $P5 \rightarrow P7 \rightarrow P6 \rightarrow P5$, $P8 \rightarrow P9 \rightarrow P8$ y $P10 \rightarrow P12 \rightarrow P11 \rightarrow P10$.

Verificamos la inclusión del nodo 1.1 en la cola (Paso 6) y comprobamos que hay que incluirlo.

Problema 1.2

Ejecutamos del paso 4 al 6. Recordemos que para el Problema de asignación 1.2 tomamos

$$I_{1.2} = \{(P6, P7)\}$$

$$E_{1.2} = \{(P7, P6)\}$$

y lo resolvemos (Paso 4), para el cual obtenemos $z_{1.2} = 200$ y la siguiente asignación

$$D1 \rightarrow P1 \rightarrow D1$$

$$D2 \rightarrow P2 \rightarrow D2$$

$$D3 \rightarrow P3 \rightarrow D3$$

$$D4 \rightarrow P4 \rightarrow D4$$

$$P5 \rightarrow P6 \rightarrow P7 \rightarrow P5$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$$

No es una solución factible ya que contiene tres subtours ilegales (Paso 5), $P5 \rightarrow P6 \rightarrow P7 \rightarrow P5$, $P8 \rightarrow P9 \rightarrow P8$ y $P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$.

Verificamos la inclusión del nodo 1.2 en la cola (Paso 6) y comprobamos que hay que incluirlo.

Paso 7: no se elimina ningún nodo.

Y volvemos al Paso 1. Seleccionamos el nodo 1.1 o el 1.2 ya que los dos proporcionan la misma función objetivo (200). Seleccionamos por el ejemplo el 1.1.

Partimos por el $P8 \rightarrow P9 \rightarrow P8$ ya que es el que tiene menor número de aristas libres (Paso 2). Examinados los puntos 1, 2 y 3 de la regla de partición obtenemos los dos siguientes subproblemas.

- **Problema 1.1.1** $\rightarrow I_{1.1.1} = \emptyset$ y $E_{1.1.1} = \{(P6, P7), (P8, P9)\}$

- **Problema 1.1.2** $\rightarrow I_{1.1.2} = \{(P8, P9)\}$ y $E_{1.1.2} = \{(P6, P7), (P9, P8)\}$

Ejecutamos del paso 4 al 6 para el problema 1.1.1 y para el problema 1.1.2.

Problema 1.1.1

Dados los conjuntos

$$I_{1.1.1} = \emptyset$$

$$E_{1.1.1} = \{(P6, P7), (P8, P9)\}$$

Resolvemos el problema de asignación correspondiente (Paso 4). Para el cual obtenemos una solución de $z_{1.1.1} = 204$ y una asignación de

$$D1 \rightarrow P3 \rightarrow D1$$

$$D2 \rightarrow P4 \rightarrow D2$$

$$\begin{aligned}
D3 &\rightarrow P2 \rightarrow D3 \\
D4 &\rightarrow P1 \rightarrow D4 \\
P5 &\rightarrow P7 \rightarrow P6 \rightarrow P5 \\
P8 &\rightarrow P10 \rightarrow P9 \rightarrow P8 \\
P11 &\rightarrow P12 \rightarrow P11
\end{aligned}$$

Solución no factible (Paso 5) ya que contiene tres subtours ilegales $P5 \rightarrow P7 \rightarrow P6 \rightarrow P5$, $P8 \rightarrow P10 \rightarrow P9 \rightarrow P8$ y $P11 \rightarrow P12 \rightarrow P11$

Verificamos la inclusión del nodo 1.1.1 en la cola (Paso 6) y comprobamos que hay que incluirlo.

Problema 1.1.2

Dados los conjuntos

$$\begin{aligned}
I_{1.1.2} &= \{(P8, P9)\} \\
E_{1.1.2} &= \{(P6, P7), (P9, P8)\}
\end{aligned}$$

Para el que obtenemos una solución (Paso 4) de $z_{1.1.2} = 204$ y la asignación siguiente.

$$\begin{aligned}
D1 &\rightarrow P1 \rightarrow D1 \\
D2 &\rightarrow P2 \rightarrow D2 \\
D3 &\rightarrow P3 \rightarrow D3 \\
D4 &\rightarrow P4 \rightarrow D4 \\
P5 &\rightarrow P7 \rightarrow P6 \rightarrow P5 \\
P8 &\rightarrow P9 \rightarrow P11 \rightarrow P12 \rightarrow P10 \rightarrow P8
\end{aligned}$$

Solución no factible ya que contiene dos subtours ilegales (Paso 5), $P5 \rightarrow P7 \rightarrow P6 \rightarrow P5$ y $P8 \rightarrow P9 \rightarrow P11 \rightarrow P12 \rightarrow P10 \rightarrow P8$

Verificamos la inclusión del nodo 1.1.2 en la cola (Paso 6) y comprobamos que hay que incluirlo.

Paso 7: no eliminamos ningún nodo. Y volvemos al paso 1. Seleccionamos el nodo 1.2 porque proporciona mejor función objetivo que el 1.1.1 y el 1.1.2.

Paso 2: Partimos por el que tenga el menor número de aristas libres, en este caso podemos partir tanto por el primero como por el segundo, ya que los dos contienen dos aristas libres porque el subtour $P5 \rightarrow P6 \rightarrow P7 \rightarrow P5$ tiene la arista $(P6, P7)$ incluida. Elegimos el subtour $P5 \rightarrow P6 \rightarrow P7 \rightarrow P5$ para partir, ya que tiene una arista incluida y así formamos cadenas con las que más tarde se podrán excluir más aristas. Examinados los tres puntos de la regla de partición obtenemos los dos siguientes subproblemas:

- **Problema 1.2.1** $\rightarrow I_{1.2.1} = \{(P6, P7)\}$ y $E_{1.2.1} = \{(P7, P6), (P5, P6)\}$
- **Problema 1.2.2** $\rightarrow I_{1.2.2} = \{(P6, P7), (P5, P6)\}$ y $E_{1.2.2} = \{(P7, P6), (P7, P5), (P8, P5), (P9, P5), (P7, P8), (P7, P9)\}$. Excluyendo los cuatro últimos arcos ya que los arcos incluidos tienen una demanda de $1700 + 1400 + 1200 = 4300$, por lo tanto no se pueden unir con ningún cliente o cadena de clientes que supere una demanda de 1700, es decir, con los nodos $P8$ y $P9$ que tienen una demanda de 1900 y 1800 respectivamente (Punto 2 regla de partición).

Ejecutamos del paso 4 al 6 para ambos problemas.

Problema 1.2.1

Dados los conjuntos

$$I_{1.2.1} = \{(P6, P7)\}$$

$$E_{1.2.1} = \{(P7, P6), (P5, P6)\}$$

Resolvemos el problema de asignación correspondiente (Paso 4), para el que obtenemos una solución de $z_{1.2.1} = 204$ y la siguiente asignación

$$D1 \rightarrow P4 \rightarrow P3 \rightarrow D2 \rightarrow P6 \rightarrow P7 \rightarrow P5 \rightarrow D1$$

$$D3 \rightarrow P2 \rightarrow D3$$

$$D4 \rightarrow P1 \rightarrow D4$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$$

Solución no factible ya que contiene dos subtours ilegales (Paso 5), $P8 \rightarrow P9 \rightarrow P8$ y $P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$

Verificamos la inclusión del nodo 1.2.1 en la cola (Paso 6) y comprobamos que hay que incluirlo.

Problema 1.2.2

Dados los conjuntos

$$I_{1.2.2} = \{(P6, P7), (P5, P6)\}$$

$$E_{1.2.2} = \{(P7, P6), (P7, P5), (P8, P5), (P9, P5), (P7, P8), (P7, P9)\}$$

Resolvemos el problema de asignación (Paso 4) para el que obtenemos una solución de $z_{1.2.2} = 210$ y la asignación

$$D1 \rightarrow P2 \rightarrow D2 \rightarrow P3 \rightarrow P4 \rightarrow D3 \rightarrow P5 \rightarrow P6 \rightarrow P7 \rightarrow D1$$

$$D4 \rightarrow P1 \rightarrow D4$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$$

Solución no factible (Paso 5) ya que contiene dos subtours ilegales, $P8 \rightarrow P9 \rightarrow P8$ y $P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$

Verificamos la inclusión del nodo 1.2.2 en la cola (Paso 6) y comprobamos que hay que incluirlo.

Paso 7: no eliminamos ningún nodo. Y volvemos al Paso 1. Como $z_{1.1.1} = 204$, $z_{1.1.2} = 204$, $z_{1.2.1} = 204$ y $z_{1.2.2} = 210$ deberíamos ramificar por el nodo que tenga menor valor de la función objetivo, en este caso daría igual ramificar tanto por el nodo 1.1.1, 1.1.2 como por el nodo 1.2.1, a partir de aquí ramificaremos por el nodo más conveniente que nos haga llegar antes a la solución óptima. Es este caso por el nodo 1.1.2. El primero de los subtours ilegales tiene tres aristas libres y el segundo cuatro ya que la arista $(P8, P9)$ está incluida. Por lo tanto ramificamos por el subtour $P5 \rightarrow P7 \rightarrow P6 \rightarrow P5$, obteniendo así tres subproblemas.

- **Problema 1.1.2.1** $\rightarrow I_{1.1.2.1} = \{(P8, P9)\}$ y $E_{1.1.2.1} = \{(P6, P7), (P9, P8), (P5, P7)\}$
- **Problema 1.1.2.2** $\rightarrow I_{1.1.2.2} = \{(P8, P9), (P5, P7)\}$ y $E_{1.1.2.2} = \{(P6, P7), (P9, P8), (P7, P6), (P7, P8), (P9, P5), (P7, P5)\}$. Excluyendo el último arco para no crear subtours ilegales (Punto 3 regla de partición) y los dos anteriores ya que los arcos incluidos tienen una demanda de $1800 + 1900 = 3700$ y $1700 + 1200 = 2900$ respectivamente, por lo tanto no se pueden unir entre ellos y debemos eliminar los arcos que los unen (Punto 2 regla de partición).
- **Problema 1.1.2.3** $\rightarrow I_{1.1.2.3} = \{(P8, P9), (P5, P7), (P7, P6)\}$ y $E_{1.1.2.3} = \{(P6, P7), (P9, P8), (P6, P5), (P6, P8), (P9, P5)\}$. Excluyendo los dos últimos arcos ya que los arcos incluidos tienen una demanda de $1800 + 1900 = 3700$ y $1700 + 1200 + 1400 = 4300$ respectivamente, por lo tanto no se pueden unir entre ellos y debemos eliminar los arcos que los unen (Punto 2 regla de partición).

Ejecutaríamos ahora del paso 4 al 6 para los tres subproblemas anteriores.

Una vez visto el funcionamiento del algoritmo, veamos algunos pasos todavía no vistos a los que se debería llegar si se desarrollase el árbol al completo.

En el problema 1.1.2.1.1.3 veremos el incumplimiento del Paso 6, es decir, cuando no se añade el nodo a la cola.

En el problema 1.1.2.1.1.2.2.3 veremos que al ramificar ejecutamos el Punto 1 de la regla de partición (réplicas de arcos).

En el problema 1.1.2.1.1.2.2.3.3 veremos como la función objetivo supera la cota inferior y por lo tanto no hay que seguir ramificando ese nodo.

Y por último en el problema 1.1.2.1.1.2.2.3.1.4.2.2 veremos como llegamos a la solución óptima.

Problema 1.1.2.1.1.3

Sean los conjuntos incluidos y excluidos:

$$I_{1.1.2.1.1.3} = \{(P8, P9), (P6, P8), (P9, P11)\}$$

$$E_{1.1.2.1.1.3} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P11, P12), (Ciudad, P6), (P11, Ciudad)\}$$

La notación $(Ciudad, P6)$ la usaremos como abreviatura para referimos a cualquier unión entre las ciudades y el nodo 6, es decir, cualquiera que no sea un depósito.

Resolvemos el correspondiente problema de asignación (Paso 4) para el que obtenemos la siguiente solución $z_{1.1.2.1.1.3}$ y la asignación

$$D1 \rightarrow P2 \rightarrow D1$$

$$D2 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow P11 \rightarrow D4 \rightarrow P1 \rightarrow D2$$

$$D3 \rightarrow P5 \rightarrow D3$$

$$P3 \rightarrow P4 \rightarrow P3$$

$$P7 \rightarrow P10 \rightarrow P12 \rightarrow P7$$

Solución no factible ya que contiene tres subtours ilegales, $D2 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow P11 \rightarrow D4$, $P3 \rightarrow P4 \rightarrow P3$ y $P7 \rightarrow P10 \rightarrow P12 \rightarrow P7$. El primero de ellos por superar

la capacidad del vehículo la demanda. Y los otros dos por no pasar por el depósito (Paso 5).

Verificamos la inclusión del nodo 1.1.2.1.1.3 en la cola (Paso 6). Comenzamos por el primer subtour ilegal.

$$\gamma(S_k) \leq |S_k| - |(S_k x S_k) \cap I_j| \quad \forall k \Rightarrow 2 \leq 4 - 3 = 1$$

Por lo tanto como no se cumple la desigualdad anterior no añadimos el nodo 1.1.2.1.1.3 a la cola, ya que los arcos incluidos superan la capacidad del camión. Esta desigualdad no siempre funciona en el primer problema, sino que al ir ramificando se terminará excluyendo dicho nodo de la cola, por eso propongo modificar el algoritmo y comprobar antes de resolver el problema que las cadenas de arcos incluidos no exceden la capacidad del camión, y si fuese así resolverlo y sino desechar dicho nodo.

Problema 1.1.2.1.1.2.2.3

Sean

$$I_{1.1.2.1.1.2.2.3} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12)\}$$

$$E_{1.1.2.1.1.2.2.3} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad)\}$$

Dicho problema de asignación proporciona la siguiente solución $z_{1.1.2.1.1.2.2.3} = 282$ y la asignación

$$D1 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow D4 \rightarrow P1 \rightarrow D2 \rightarrow P7 \rightarrow P10 \rightarrow P11 \rightarrow P12 \rightarrow D1$$

$$D3 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow P5 \rightarrow D3$$

Solución no factible ya que contiene un subtour ilegal $D3 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow P5 \rightarrow D3$, este subtour es ilegal porque la demanda entre dos depósitos excede la capacidad del camión ($P2, P3, P4$ y $P5$, $1700 + 1500 + 1400 + 1700 = 6300 > 6000$). Que tiene cuatro aristas libres ya que la $(P3, P4)$ está incluida. Daremos sólo el primer subproblema puesto que es en el que aplicaremos el punto 1 de la regla de partición.

- **Problema 1.1.2.1.1.2.2.3.1** $\rightarrow I_{1.1.2.1.1.2.2.3.1} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12)\}$ y $E_{1.1.2.1.1.2.2.3.1} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad), (D3, P2), (D1, P2), (D2, P2), (D4, P2)\}$. Excluyendo los tres últimos arcos por ser réplica del anterior (Punto 1 regla de partición).

Problema 1.1.2.1.1.2.2.3.3

Dados los conjuntos

$$I_{1.1.2.1.1.2.2.3.3} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12), (D3, P2), (P2, P3)\}$$

$$E_{1.1.2.1.1.2.2.3.3} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad), (P4, P5)\}$$

Problema que proporciona una solución de $z_{1.1.2.1.1.2.2.3.3} = 298$, y como $z_{1.1.2.1.1.2.2.3.3} > z^*$ desecharmos este nodo y no seguimos ramificando por él.

Problema 1.1.2.1.1.2.2.2.3.1.4.2.2

Sean

$$I_{1.1.2.1.1.2.2.2.3.1.4.2.2} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12), \\ (D1, P6), (P9, D4), (D4, P1), (D2, P5), (P2, P3)\}$$

$$E_{1.1.2.1.1.2.2.2.3.1.4.2.2} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), \\ (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad), (D3, P2), \\ (D1, P2), (D2, P2), (D4, P2), (P1, D3), (P1, D1), (P1, D2), (P1, D4), (P5, P2), (P4, P2)\}$$

Y por último resolviendo este problema de asignación obtenemos $z_{1.1.2.1.1.2.2.2.3.1.4.2.2} = 290$ con la siguiente asignación

$$D1 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow D4 \rightarrow P1 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow D3 \rightarrow P7 \rightarrow$$

$$\rightarrow P10 \rightarrow P11 \rightarrow P12 \rightarrow D1$$

$$D2 \rightarrow P5 \rightarrow D2$$

Solución factible ya que no contiene ningún subtour ilegal, y como sabemos por el artículo de Dantzig y Ramser solución óptima. Cabe señalar que este problema tiene solución múltiple, ya que la asignación

$$D1 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow D4 \rightarrow P1 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow D3 \rightarrow P7 \rightarrow$$

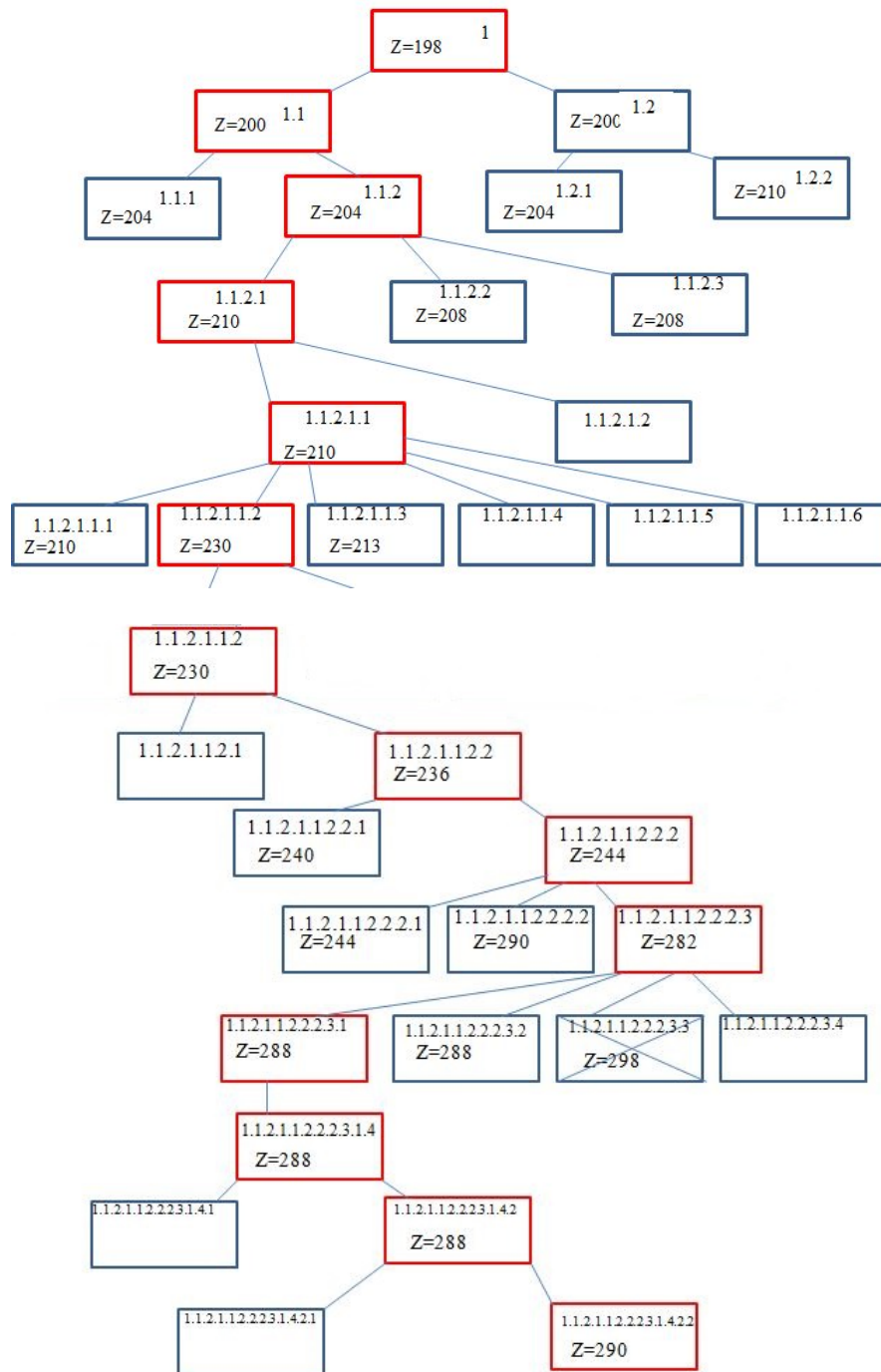
$$\rightarrow P12 \rightarrow P11 \rightarrow P10 \rightarrow D1$$

$$D2 \rightarrow P5 \rightarrow D2$$

también es válida y su función objetivo tiene el mismo valor (290).

Aunque el árbol que habría que expandir para solucionar el problema sería muy grande, en la imagen que tenemos a continuación podemos ver la parte del árbol que hemos expandido para solucionar el problema. Los nodos marcados en rojo son los que conducen a la solución óptima.

Anteriormente se ha seguido el algoritmo paso a paso para la resolución de los problemas 1, 1.1, 1.2, 1.1.1, 1.1.2, 1.2.1 y 1.2.2. A partir de este punto hemos solucionado problemas específicos en los que se incluían partes del algoritmo no vistas anteriormente. Se puede comprobar cual es la correspondencia entre nodo y problema con el número que encontramos en la esquina superior derecha de los recuadros.



Cabe señalar que este ejemplo se resolvió en primer lugar con el programa WinQSB, pero se encontró un error de software y nos vimos obligados a resolverlo de nuevo con Lingo.

Bibliografía

- [1] PAOLO TOTH Y DANIELE VIGO: *Vehicle Routing. Problems, Methods and Applications. Second Edition.* Society for Industrial and Applied Mathematics and the Mathematical Optimization Society. (2014)
- [2] WILLIAM J. COOK: *In pursuit of the Travelling Salesman.* Princeton University Press. (2012)
- [3] GILBERT LAPORTE: *The Vehicle Routing Problem: An overview of exact and approximate algorithms.* European Journal of Operational Research . 59. 345-358 (1992)
- [4] GREGORY GUTIN AND ABRAHAM P. PUNNEN: *The Travelling Salesman Problem and Its Variations.* Springer. (2007)
- [5] PAOLO TOTH, DANIELE VIGO: *Models, relaxations and exact approaches for the capacitated vehicle routing problem.* Discrete Applied Mathematics. 123. Elsevier. 487-512 (2002)
- [6] G. B. DANTZIG AND J. H. RAMSER: *The Truck Dispatching Problem.* Management Science. 6. Informs. 80-91 (1959)
- [7] G. B. DANTZIG, R. FULKERSON AND J. JOHNSON: *Solution of a Large-Scale Traveling Salesman Problem.* J. Opns. Res. Soc. Am. 2. 393-410 (1954)
- [8] DANTZIG, G. B.: *Discrete-Variable Extremum Problems.* J. Opns. Res. Soc. Am. 5. 266-277 (1957)
- [9] GOMORY, R.: *Essentials of an Algorithm for Integer Solutions to Linear Programs.* (communicated to Bull. Amer. Math. Society in a letter from Princeton) (1958)
- [10] G. B. DANTZIG: *Maximization of a Linear Function of Variables Subject to Linear Inequalities.* Chapter XXI in “Activity Analysis of Production and Allocation”, edited by T. C. Koopmans, Cowles Commission Monograph Nr. 13, .John Wiley & Sons, New York, 1951.
- [11] G. CLARKE AND J. W. WRIGHT: *Scheduling of Vehicles from a Central Depot to a Number of Delivery Points.* JOperations Research 12. 568-571 (1964)
- [12] J. D. C. LITTLE, K. G. MURTY, D. W. SWEENEY AND C. KAREL : *An algorithm for the travelling salesman problem.* Operational Research Society 11. 979 (1963)
- [13] N. CHRISTOFIDES AND S. EILON: *An Algorithm for the Vehicle-dispatching Problem.* Operational Research Society 20. 309-318 (1969)

- [14] GILBERT LAPORTE, HÉLÈNE MERCURE AND YVES NOBERT: *An Exact Algorithm for the Asymmetrical Capacited Vehicle Routing Problem*. Networks 16. 33-46 (1986)
- [15] G. B. DANTZIG, D. R. FULKERSON AND S. M. JOHNSON: *On a linear-programming, combinatorial approach to the travelling-salesman problem*. Operation Research (1959)

On a linear-programming, combinatorial approach to the travelling-salesman problem”, en Operation Research

Anexo

En este anexo explicaremos paso a paso la solución del problema planteado en el capítulo 5, esta solución será una solución guiada para llegar antes la solución óptima, ya que es conocida por el artículo de Dantzig y Ramser. Por completitud se repite el problema desde el principio, aunque los primeros pasos ya se hayan desarrollado en el capítulo 5.

Problema 1

Paso 0: Sea $z^* = 294$ obtenido por el algoritmo heurístico de Dantzig y Ramser, que se ha estudiado anteriormente. Tomamos los conjuntos $I_1 = \emptyset = E_1$ y resolvemos el primer problema de asignación para el que hemos usado WinQSB y Lingo, cabe señalar que el ejemplo que estamos resolviendo no es dirigido, de ahí que la matriz sea simétrica, y que las entradas que contienen un 1000 se han tomado de esa manera para penalizar dichos costos y que no se puedan asignar dichos nodos.

From \ To	deposito1	deposito2	deposito3	deposito4	nodo1	nodo2	nodo3	nodo4	nodo5	nodo6	nodo7	nodo8	nodo9	nodo10	nodo11	nodo12
deposito1	1000	1000	1000	1000	9	14	21	23	22	25	32	36	38	42	50	52
deposito2	1000	1000	1000	1000	9	14	21	23	22	25	32	36	38	42	50	52
deposito3	1000	1000	1000	1000	9	14	21	23	22	25	32	36	38	42	50	52
deposito4	1000	1000	1000	1000	9	14	21	23	22	25	32	36	38	42	50	52
nodo1	9	9	9	9	1000	5	12	22	21	24	31	35	37	41	49	51
nodo2	14	14	14	14	5	1000	7	17	16	23	26	30	36	36	44	46
nodo3	21	21	21	21	12	7	1000	10	21	30	27	37	43	31	37	39
nodo4	23	23	23	23	22	17	10	1000	19	28	25	35	41	29	31	29
nodo5	22	22	22	22	21	16	21	19	1000	9	10	16	22	20	28	30
nodo6	25	25	25	25	24	23	30	28	9	1000	7	11	13	17	25	27
nodo7	32	32	32	32	31	26	27	25	10	7	1000	10	16	10	18	20
nodo8	36	36	36	36	35	30	37	35	16	11	10	1000	6	6	14	16
nodo9	38	38	38	38	37	36	43	41	22	13	16	6	1000	12	12	20
nodo10	42	42	42	42	41	36	31	29	20	17	10	6	12	1000	8	10
nodo11	50	50	50	50	49	44	37	31	28	25	18	14	12	8	1000	10
nodo12	52	52	52	52	51	46	39	29	30	27	20	16	20	10	10	1000

Para este problema obtenemos la siguiente solución:

$$D1 \rightarrow P1 \rightarrow D2 \rightarrow P2 \rightarrow D3 \rightarrow P3 \rightarrow P4 \rightarrow D4 \rightarrow P5 \rightarrow D1$$

$$P6 \rightarrow P7 \rightarrow P6$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$$

Con $z_1 = 198$.

La solución obtenida no es factible ya que contiene tres subtours ilegales, ya que no pasan por el depósito, por lo tanto añadimos el nodo 1 a la cola.

$$P6 \rightarrow P7 \rightarrow P6$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$$

Como la cola no está vacía seleccionamos el nodo 1 (Paso 1). Particionamos por el que tenga menor número de arcos, sin tener en cuenta los incluidos (Paso 2). Es decir, el primero y el segundo tienen dos aristas libres y el tercero tres. Partimos por cualquiera de los dos primeros, elegimos el primero.

Examinados los tres puntos de la regla de partición obtenemos dos subproblemas, con sus correspondientes conjuntos de incluidos y excluidos:

- **Problema 1.1** $\rightarrow I_{1.1} = \emptyset$ y $E_{1.1} = \{(P6, P7)\}$
- **Problema 1.2** $\rightarrow I_{1.2} = \{(P6, P7)\}$ y $E_{1.2} = \{(P7, P6)\}$

Para cada subproblema ejecutamos los pasos 4 al 6.

Problema 1.1

Para el Problema de asignación 1.1 tomamos

$$I_{1.1} = \emptyset$$

$$E_{1.1} = \{(P6, P7)\}$$

y lo resolvemos (Paso 4), para el cual obtenemos $z_{1.1} = 200$ y la siguiente asignación

$$D1 \rightarrow P1 \rightarrow D1$$

$$D2 \rightarrow P2 \rightarrow D2$$

$$D3 \rightarrow P3 \rightarrow D3$$

$$D4 \rightarrow P4 \rightarrow D4$$

$$P5 \rightarrow P7 \rightarrow P6 \rightarrow P5$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P12 \rightarrow P11 \rightarrow P10$$

Solución que no es factible porque contiene tres subtours ilegales (Paso 5).

$$P5 \rightarrow P7 \rightarrow P6 \rightarrow P5$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P12 \rightarrow P11 \rightarrow P10$$

Verificamos la inclusión del nodo 1.1 en la cola (Paso 6) y comprobamos que hay que incluirlo.

Problema 1.2

Ejecutamos del paso 4 al 6. Recordemos que para el Problema de asignación 1.2 tomamos

$$I_{1.2} = \{(P6, P7)\}$$

$$E_{1.2} = \{(P7, P6)\}$$

y lo resolvemos (Paso 4), para el cual obtenemos $z_{1.2} = 200$ y la siguiente asignación

$$D1 \rightarrow P1 \rightarrow D1$$

$$D2 \rightarrow P2 \rightarrow D2$$

$$D3 \rightarrow P3 \rightarrow D3$$

$$D4 \rightarrow P4 \rightarrow D4$$

$$P5 \rightarrow P6 \rightarrow P7 \rightarrow P5$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$$

No es una solución factible ya que contiene tres subtours ilegales (Paso 5)

$$P5 \rightarrow P6 \rightarrow P7 \rightarrow P5$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$$

Verificamos la inclusión del nodo 1.2 en la cola (Paso 6) y comprobamos que hay que incluirlo.

Paso 7: no se elimina ningún nodo.

Y volvemos al Paso 1. Seleccionamos el nodo 1.1 o el 1.2 ya que los dos proporcionan la misma función objetivo (200). Seleccionamos por el ejemplo el 1.1.

Partimos por el $P8 \rightarrow P9 \rightarrow P8$ ya que es el que tiene menor número de aristas libres (Paso 2). Examinados los puntos 1, 2 y 3 de la regla de partición obtenemos los dos siguientes subproblemas.

- **Problema 1.1.1** $\rightarrow I_{1.1.1} = \emptyset$ y $E_{1.1.1} = \{(P6, P7), (P8, P9)\}$

- **Problema 1.1.2** $\rightarrow I_{1.1.2} = \{(P8, P9)\}$ y $E_{1.1.2} = \{(P6, P7), (P9, P8)\}$

Ejecutamos del paso 4 al 6 para el problema 1.1.1 y para el problema 1.1.2.

Problema 1.1.1

Dados los conjuntos

$$I_{1.1.1} = \emptyset$$

$$E_{1.1.1} = \{(P6, P7), (P8, P9)\}$$

Resolvemos el problema de asignación correspondiente (Paso 4). Para el cual obtenemos una solución de $z_{1.1.1} = 204$ y una asignación de

$$D1 \rightarrow P3 \rightarrow D1$$

$$D2 \rightarrow P4 \rightarrow D2$$

$$D3 \rightarrow P2 \rightarrow D3$$

$$D4 \rightarrow P1 \rightarrow D4$$

$$P5 \rightarrow P7 \rightarrow P6 \rightarrow P5$$

$$P8 \rightarrow P10 \rightarrow P9 \rightarrow P8$$

$$P11 \rightarrow P12 \rightarrow P11$$

Solución no factible (Paso 5) ya que contiene tres subtours ilegales

$$P5 \rightarrow P7 \rightarrow P6 \rightarrow P5$$

$$P8 \rightarrow P10 \rightarrow P9 \rightarrow P8$$

$$P11 \rightarrow P12 \rightarrow P11$$

Verificamos la inclusión del nodo 1.1.1 en la cola (Paso 6) y comprobamos que hay que incluirlo.

Problema 1.1.2

Dados los conjuntos

$$I_{1.1.2} = \{(P8, P9)\}$$

$$E_{1.1.2} = \{(P6, P7), (P9, P8)\}$$

Para el que obtenemos una solución (Paso 4) de $z_{1.1.2} = 204$ y la asignación siguiente.

$$D1 \rightarrow P1 \rightarrow D1$$

$$D2 \rightarrow P2 \rightarrow D2$$

$$D3 \rightarrow P3 \rightarrow D3$$

$$D4 \rightarrow P4 \rightarrow D4$$

$$P5 \rightarrow P7 \rightarrow P6 \rightarrow P5$$

$$P8 \rightarrow P9 \rightarrow P11 \rightarrow P12 \rightarrow P10 \rightarrow P8$$

Solución no factible ya que contiene dos subtours ilegales (Paso 5)

$$P5 \rightarrow P7 \rightarrow P6 \rightarrow P5$$

$$P8 \rightarrow P9 \rightarrow P11 \rightarrow P12 \rightarrow P10 \rightarrow P8$$

Verificamos la inclusión del nodo 1.1.2 en la cola (Paso 6) y comprobamos que hay que incluirlo.

Paso 7: no eliminamos ningún nodo. Y volvemos al paso 1. Seleccionamos el nodo 1.2 porque proporciona mejor función objetivo que el 1.1.1 y el 1.1.2.

Paso 2: Partimos por el que tenga el menor número de aristas libres, en este caso podemos partir tanto por el primero como por el segundo, ya que los dos contienen dos aristas libres porque el subtour $P5 \rightarrow P6 \rightarrow P7 \rightarrow P5$ tiene la arista $(P6, P7)$ incluida. Elegimos el subtour $P5 \rightarrow P6 \rightarrow P7 \rightarrow P5$ para partir, ya que tiene una arista incluida y así formamos cadenas con las que más tarde se podrán excluir más aristas. Examinados los tres puntos de la regla de partición obtenemos los dos siguientes subproblemas:

- **Problema 1.2.1** $\rightarrow I_{1.2.1} = \{(P6, P7)\}$ y $E_{1.2.1} = \{(P7, P6), (P5, P6)\}$
- **Problema 1.2.2** $\rightarrow I_{1.2.2} = \{(P6, P7), (P5, P6)\}$ y $E_{1.2.2} = \{(P7, P6), (P7, P5), (P8, P5), (P9, P5), (P7, P8), (P7, P9)\}$. Excluyendo los cuatro últimos arcos ya que los arcos incluidos tienen una demanda de $1700 + 1400 + 1200 = 4300$, por lo tanto no se pueden unir con ningún cliente o cadena de clientes que supere una demanda de 1700, es decir, con los nodos $P8$ y $P9$ que tienen una demanda de 1900 y 1800 respectivamente (Punto 2 regla de partición).

Ejecutamos del paso 4 al 6 para ambos problemas.

Problema 1.2.1

Dados los conjuntos

$$I_{1.2.1} = \{(P6, P7)\}$$

$$E_{1.2.1} = \{(P7, P6), (P5, P6)\}$$

Resolvemos el problema de asignación correspondiente (Paso 4), para el que obtenemos una solución de $z_{1.2.1} = 204$ y la siguiente asignación

$$D1 \rightarrow P4 \rightarrow P3 \rightarrow D2 \rightarrow P6 \rightarrow P7 \rightarrow P5 \rightarrow D1$$

$$D3 \rightarrow P2 \rightarrow D3$$

$$D4 \rightarrow P1 \rightarrow D4$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$$

Solución no factible ya que contiene dos subtours ilegales (Paso 5)

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$$

Verificamos la inclusión del nodo 1.2.1 en la cola (Paso 6) y comprobamos que hay que incluirlo.

Problema 1.2.2

Dados los conjuntos

$$I_{1.2.2} = \{(P6, P7), (P5, P6)\}$$

$$E_{1.2.2} = \{(P7, P6), (P7, P5), (P8, P5), (P9, P5), (P7, P8), (P7, P9)\}$$

Resolvemos el problema de asignación (Paso 4) para el que obtenemos una solución de $z_{1.2.2} = 210$ y la asignación

$$D1 \rightarrow P2 \rightarrow D2 \rightarrow P3 \rightarrow P4 \rightarrow D3 \rightarrow P5 \rightarrow P6 \rightarrow P7 \rightarrow D1$$

$$D4 \rightarrow P1 \rightarrow D4$$

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$$

Solución no factible (Paso 5) ya que contiene dos subtours ilegales

$$P8 \rightarrow P9 \rightarrow P8$$

$$P10 \rightarrow P11 \rightarrow P12 \rightarrow P10$$

Verificamos la inclusión del nodo 1.2.2 en la cola (Paso 6) y comprobamos que hay que incluirlo.

Paso 7: no eliminamos ningún nodo. Y volvemos al Paso 1. Como $z_{1.1.1} = 204$, $z_{1.1.2} = 204$, $z_{1.2.1} = 204$ y $z_{1.2.2} = 210$ deberíamos ramificar por el nodo que tenga menor valor de la función objetivo, en este caso daría igual ramificar tanto por el nodo 1.1.1, 1.1.2 como por el nodo 1.2.1, a partir de aquí ramificaremos por el nodo más conveniente que nos haga llegar antes a la solución óptima. Es este caso por el nodo 1.1.2. El primero de los subtours ilegales tiene tres aristas libres y el segundo cuatro ya que la arista $(P8, P9)$ está incluida. Por lo tanto ramificamos por el subtour $P5 \rightarrow P7 \rightarrow P6 \rightarrow P5$, obteniendo así tres subproblemas.

- **Problema 1.1.2.1** $\rightarrow I_{1.1.2.1} = \{(P8, P9)\}$ y $E_{1.1.2.1} = \{(P6, P7), (P9, P8), (P5, P7)\}$
- **Problema 1.1.2.2** $\rightarrow I_{1.1.2.2} = \{(P8, P9), (P5, P7)\}$ y $E_{1.1.2.2} = \{(P6, P7), (P9, P8), (P7, P6), (P7, P8), (P9, P5), (P7, P5)\}$. Excluyendo el último arco para no crear subtours ilegales (Punto 3 regla de partición) y los dos anteriores ya que los arcos incluidos tienen una demanda de $1800 + 1900 = 3700$ y $1700 + 1200 = 2900$ respectivamente, por lo tanto no se pueden unir entre ellos y debemos eliminar los arcos que los unen (Punto 2 regla de partición).
- **Problema 1.1.2.3** $\rightarrow I_{1.1.2.3} = \{(P8, P9), (P5, P7), (P7, P6)\}$ y $E_{1.1.2.3} = \{(P6, P7), (P9, P8), (P6, P5), (P6, P8), (P9, P5)\}$. Excluyendo los dos últimos arcos ya que los arcos incluidos tienen una demanda de $1800 + 1900 = 3700$ y $1700 + 1200 + 1400 = 4300$ respectivamente, por lo tanto no se pueden unir entre ellos y debemos eliminar los arcos que los unen (Punto 2 regla de partición).

Ejecutamos del paso 4 al 6 para los tres subproblemas anteriores.

Problema 1.1.2.1

Sean

$$I_{1.1.2.1} = \{(P8, P9)\}$$

$$E_{1.1.2.1} = \{(P6, P7), (P9, P8), (P5, P7)\}$$

Para el que obtenemos la siguiente solución $z_{1.1.2.1} = 210$.

$$D1 \rightarrow P3 \rightarrow D2 \rightarrow P4 \rightarrow D1$$

$$D3 \rightarrow P2 \rightarrow D3$$

$$D4 \rightarrow P1 \rightarrow D4$$

$$P5 \rightarrow P6 \rightarrow P5$$

$$P7 \rightarrow P8 \rightarrow P9 \rightarrow P11 \rightarrow P12 \rightarrow P10 \rightarrow P7$$

Solución no factible ya que contiene dos subtours ilegales

$$P5 \rightarrow P6 \rightarrow P5$$

$$P7 \rightarrow P8 \rightarrow P9 \rightarrow P11 - P12 - P10 - P7$$

Partimos por el $P5 \rightarrow P6 \rightarrow P5$ y obtenemos dos subproblemas:

- **Problema 1.1.2.1.1** $\rightarrow I_{1.1.2.1.1} = \{(P8, P9)\}$ y $E_{1.1.2.1.1} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6)\}$
- **Problema 1.1.2.1.2** $\rightarrow I_{1.1.2.1.2} = \{(P8, P9), (P5, P6)\}$ y $E_{1.1.2.1.2} = \{(P6, P7), (P9, P8), (P5, P7), (P6, P5), (P6, P8), (P9, P5)\}$

Ahora ramificaremos por el nodo 1.1.2.1.1

Problema 1.1.2.1.1

Sean

$$I_{1.1.2.1.1} = \{(P8, P9)\}$$

$$E_{1.1.2.1.1} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6)\}$$

Dicho problema de asignación proporciona una solución de $z_{1.1.2.1.1} = 210$ y la asignación

$$D1 \rightarrow P5 \rightarrow D1$$

$$D2 \rightarrow P2 \rightarrow D2$$

$$D3 \rightarrow P4 \rightarrow P3 \rightarrow D3$$

$$D4 \rightarrow P1 \rightarrow D4$$

$$P6 \rightarrow P8 \rightarrow P9 \rightarrow P11 \rightarrow P12 \rightarrow P10 \rightarrow P7 \rightarrow P6$$

Solución no factible ya que el subtour $P6 \rightarrow P8 \rightarrow P9 \rightarrow P11 \rightarrow P12 \rightarrow P10 \rightarrow P7 \rightarrow P6$ es ilegal porque no pasa por el depósito.

Ramificamos por este subtour que contiene 6 aristas libres ya que la arista $(P8, P9)$ está incluida. Examinando los tres puntos de la regla de partición obtenemos los siguientes 6 subproblemas:

- **Problema 1.1.2.1.1.1** $\rightarrow I_{1.1.2.1.1.1} = \{(P8, P9)\}$ y $E_{1.1.2.1.1.1} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P6, P8)\}$
- **Problema 1.1.2.1.1.2** $\rightarrow I_{1.1.2.1.1.2} = \{(P8, P9), (P6, P8)\}$ y $E_{1.1.2.1.1.2} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6)\}$.
- **Problema 1.1.2.1.1.3** $\rightarrow I_{1.1.2.1.1.3} = \{(P8, P9), (P6, P8), (P9, P11)\}$ y $E_{1.1.2.1.1.3} = \{(P6, P7), (P9, P8), (P5, P7), (P11, P12), (Ciudad, P6), (P11, Ciudad)\}$.
- **Problema 1.1.2.1.1.4** $\rightarrow I_{1.1.2.1.1.4} = \{(P8, P9), (P6, P8), (P9, P11), (P11, P12)\}$ y $E_{1.1.2.1.1.4} = \{(P6, P7), (P9, P8), (P5, P7), (P11, P10), (Ciudad, P6), (P12, Ciudad)\}$.
- **Problema 1.1.2.1.1.5** $\rightarrow I_{1.1.2.1.1.5} = \{(P8, P9), (P6, P8), (P9, P11), (P11, P12), (P12, P10)\}$ y $E_{1.1.2.1.1.5} = \{(P6, P7), (P9, P8), (P5, P7), (P10, P7), (Ciudad, P6), (P10, Ciudad)\}$.
- **Problema 1.1.2.1.1.6** $\rightarrow I_{1.1.2.1.1.6} = \{(P8, P9), (P6, P8), (P9, P11), (P11, P12), (P12, P10), (P10, P7)\}$ y $E_{1.1.2.1.1.6} = \{(P6, P7), (P9, P8), (P5, P7), (P7, P6), (Ciudad, P6), (P7, Ciudad)\}$.

Problema 1.1.2.1.1.2

Sean

$$I_{1.1.2.1.1.2} = \{(P8, P9), (P6, P8)\}$$

$$E_{1.1.2.1.1.2} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6)\}$$

Resolvemos el problema de asignación para el cual obtenemos $z_{1.1.2.1.1.2} = 230$ y la asignación

$$\begin{aligned} D1 &\rightarrow P2 \rightarrow D1 \\ D2 &\rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow D4 \rightarrow P1 \rightarrow D2 \\ D3 &\rightarrow P5 \rightarrow D3 \\ P3 &\rightarrow P4 \rightarrow P3 \\ P7 &\rightarrow P10 \rightarrow P7 \\ P11 &\rightarrow P12 \rightarrow P11 \end{aligned}$$

Solución no factible ya que contiene tres subtours ilegales

$$\begin{aligned} P3 &\rightarrow P4 \rightarrow P3 \\ P7 &\rightarrow P10 \rightarrow P7 \\ P11 &\rightarrow P12 \rightarrow P11 \end{aligned}$$

Ramificamos por cualquiera de los tres ya que tienen el mismo número de aristas libres.

En este caso lo haremos por $P3 \rightarrow P4 \rightarrow P3$.

Obtenemos dos subproblemas:

- **Problema 1.1.2.1.1.2.1** $\rightarrow I_{1.1.2.1.1.2.1} = \{(P8, P9), (P6, P8)\}$ y $E_{1.1.2.1.1.2.1} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P3, P4)\}$
- **Problema 1.1.2.1.1.2.2** $\rightarrow I_{1.1.2.1.1.2.2} = \{(P8, P9), (P6, P8), (P3, P4)\}$ y $E_{1.1.2.1.1.2.2} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3)\}$

Resolvemos el 1.1.2.1.1.2.2.

Problema 1.1.2.1.1.2.2

Sean

$$I_{1.1.2.1.1.2.2} = \{(P8, P9), (P6, P8), (P3, P4)\}$$

$$E_{1.1.2.1.1.2.2} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), \\ (P9, Ciudad), (Ciudad, P6), (P4, P3)\}$$

Resolvemos el problema de asignación y obtenemos $z_{1.1.2.1.1.2.2} = 236$

$$D1 \rightarrow P5 \rightarrow D3 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow D4 \rightarrow P1 \rightarrow D2 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow D1$$

$$P7 \rightarrow P10 \rightarrow P7$$

$$P11 \rightarrow P12 \rightarrow P11$$

Solución no factible ya que contiene dos subtours ilegales

$$P7 \rightarrow P10 \rightarrow P7$$

$$P11 \rightarrow P12 \rightarrow P11$$

Ramificamos por el primero y obtenemos

- **Problema 1.1.2.1.1.2.2.1** $\rightarrow I_{1.1.2.1.1.2.2.1} = \{(P8, P9), (P6, P8), (P3, P4)\}$ y $E_{1.1.2.1.1.2.2.1} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P7, P10)\}$
- **Problema 1.1.2.1.1.2.2.2** $\rightarrow I_{1.1.2.1.1.2.2.2} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10)\}$ y $E_{1.1.2.1.1.2.2.2} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7)\}$

Resolvemos el problema 1.1.2.1.1.2.2.2

Problema 1.1.2.1.1.2.2.2

Sean

$$I_{1.1.2.1.1.2.2.2} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10)\}$$

$$E_{1.1.2.1.1.2.2.2} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), \\ (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7)\}$$

Para el que se obtiene una solución de $z_{1.1.2.1.1.2.2.2} = 244$ y la asignación

$$D1 \rightarrow P5 \rightarrow D4 \rightarrow P1 \rightarrow D3 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow D2 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow D1$$

$$P7 \rightarrow P10 \rightarrow P11 \rightarrow P12 \rightarrow P7$$

Solución no factible por el subtour $P7 \rightarrow P10 \rightarrow P11 \rightarrow P12 \rightarrow P7$. Ramificamos y obtenemos 3 subproblemas ya que la arista $(P7, P10)$ está incluida.

- **Problema 1.1.2.1.1.2.2.2.1** $\rightarrow I_{1.1.2.1.1.2.2.2.1} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10)\}$ y $E_{1.1.2.1.1.2.2.2.1} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P10, P11)\}$

- **Problema 1.1.2.1.1.2.2.2.2** $\rightarrow I_{1.1.2.1.1.2.2.2.2} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11)\}$ y $E_{1.1.2.1.1.2.2.2.2} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P11, P12), (P11, P5), (P4, P7), (P11, P3)\}$
- **Problema 1.1.2.1.1.2.2.2.3** $\rightarrow I_{1.1.2.1.1.2.2.2.3} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12)\}$ y $E_{1.1.2.1.1.2.2.2.3} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad)\}$

Resolvemos el problema 1.1.2.1.1.2.2.2.3

Problema 1.1.2.1.1.2.2.2.3

Sean

$$I_{1.1.2.1.1.2.2.2.3} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12)\}$$

$$E_{1.1.2.1.1.2.2.2.3} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad)\}$$

Dicho problema de asignación proporciona la siguiente solución $z_{1.1.2.1.1.2.2.2.3} = 282$ y la asignación

$$D1 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow D4 \rightarrow P1 \rightarrow D2 \rightarrow P7 \rightarrow P10 \rightarrow P11 \rightarrow P12 \rightarrow D1$$

$$D3 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow P5 \rightarrow D3$$

Solución no factible ya que contiene un subtour ilegal $D3 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow P5 \rightarrow D3$, este subtour es ilegal porque la demanda entre dos depósitos excede la capacidad del camión ($P2, P3, P4$ y $P5$, $1700 + 1500 + 1400 + 1700 = 6300 > 6000$). Que tiene cuatro aristas libres ya que la $(P3, P4)$ está incluida. Daremos sólo el primer subproblema ya que es por el que debemos seguir ramificando.

- **Problema 1.1.2.1.1.2.2.2.3.1** $\rightarrow I_{1.1.2.1.1.2.2.2.3.1} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12)\}$ y $E_{1.1.2.1.1.2.2.2.3.1} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad), (D3, P2), (D1, P2), (D2, P2), (D4, P2)\}$. Excluyendo los tres últimos arcos por ser réplica del anterior (Punto 1 regla de partición)

Problema 1.1.2.1.1.2.2.2.3.1

Sean

$$I_{1.1.2.1.1.2.2.2.3.1} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12)\}$$

$$E_{1.1.2.1.1.2.2.2.3.1} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad), (D3, P2), (D1, P2), (D2, P2), (D4, P2)\}$$

Para el cual obtenemos un función objetivo de $z_{1.1.2.1.1.2.2.2.3.1} = 288$

$$D1 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow D4 \rightarrow P1 \rightarrow D3 \rightarrow P5 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow D2 \rightarrow \\ \rightarrow P7 \rightarrow P10 \rightarrow P11 \rightarrow P12 \rightarrow D1$$

Que es ilegal ya que las demandas entre los depósitos D3 y D2 exceden la capacidad del camión. Ramificamos y nos quedamos con el problema que nos interesa

- **Problema 1.1.2.1.1.2.2.2.3.1.4** $\rightarrow I_{1.1.2.1.1.2.2.2.3.1.4} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12), (D1, P6), (P9, D4), (D4, P1)\}$ y $E_{1.1.2.1.1.2.2.2.3.1.4} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad), (D3, P2), (D1, P2), (D2, P2), (D4, P2), (P1, D3), (P1, D1), (P1, D2), (P1, D4)\}$

Problema 1.1.2.1.1.2.2.2.3.1.4

Sean

$$I_{1.1.2.1.1.2.2.2.3.1.4} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12), \\ (D1, P6), (P9, D4), (D4, P1)\}$$

$$E_{1.1.2.1.1.2.2.2.3.1.4} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), \\ (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), \\ (P12, Ciudad), (D3, P2), (D1, P2), (D2, P2), (D4, P2), \\ (P1, D3), (P1, D1), (P1, D2), (P1, D4)\}$$

Con solución $z_{1.1.2.1.1.2.2.2.3.1.4}$

$$D1 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow D4 \rightarrow P1 \rightarrow D3 \rightarrow P7 \rightarrow P10 \rightarrow P11 \rightarrow P12 \rightarrow D1 \\ D2 \rightarrow P5 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow D2$$

Con un sbtour ilegal $D2 \rightarrow P5 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow D2$ ya que excede la capacidad del camión. Dará lugar a cuatro subproblemas ya que la arista $(P3, P4)$ está incluida. El que nos interesa es el segundo.

- **Problema 1.1.2.1.1.2.2.2.3.1.4.2** $\rightarrow I_{1.1.2.1.1.2.2.2.3.1.4.2} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12), (D1, P6), (P9, D4), (D4, P1), (D2, P5)\}$ y

$$E_{1.1.2.1.1.2.2.2.3.1.4.2} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), \\ (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad), (D3, P2), \\ (D1, P2), (D2, P2), (D4, P2), (P1, D3), (P1, D1), (P1, D2), (P1, D4), (P5, P2)\}$$

Problema 1.1.2.1.1.2.2.3.1.4.2

Sean los conjuntos

$$I_{1.1.2.1.1.2.2.3.1.4.2} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12), (D1, P6), (P9, D4), (D4, P1), (D2, P5)\}$$

$$E_{1.1.2.1.1.2.2.3.1.4.2} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad), (D3, P2), (D1, P2), (D2, P2), (D4, P2), (P1, D3), (P1, D1), (P1, D2), (P1, D4), (P5, P2)\}$$

Este problema de asignación nos da $z_{1.1.2.1.1.2.2.3.1.4.2} = 288$

$$D1 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow D4 \rightarrow P1 \rightarrow D3 \rightarrow P7 \rightarrow P10 \rightarrow P11 \rightarrow P12 \rightarrow D1$$

$$D2 \rightarrow P5 \rightarrow D2$$

$$P2 \rightarrow P3 \rightarrow P4 \rightarrow P2$$

Solución no factible ya que contiene un subtour que no pasa por el depósito. Ramificamos y nos da dos subproblemas ya que la arista $(P3, P4)$ pertenece al conjunto de los incluidos. Resolvemos el segundo que es el que nos hará llegar antes a la solución óptima.

- **Problema 1.1.2.1.1.2.2.3.1.4.2.2** $\rightarrow I_{1.1.2.1.1.2.2.3.1.4.2.2} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12), (D1, P6), (P9, D4), (D4, P1), (D2, P5), (P2, P3)\}$
y

$$E_{1.1.2.1.1.2.2.3.1.4.2.2} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad), (D3, P2), (D1, P2), (D2, P2), (D4, P2), (P1, D3), (P1, D1), (P1, D2), (P1, D4), (P5, P2), (P4, P2)\}$$

Problema 1.1.2.1.1.2.2.3.1.4.2.2

Sean

$$I_{1.1.2.1.1.2.2.3.1.4.2.2} = \{(P8, P9), (P6, P8), (P3, P4), (P7, P10), (P10, P11), (P11, P12), (D1, P6), (P9, D4), (D4, P1), (D2, P5), (P2, P3)\}$$

$$E_{1.1.2.1.1.2.2.3.1.4.2.2} = \{(P6, P7), (P9, P8), (P5, P7), (P5, P6), (P9, P11), (P9, Ciudad), (Ciudad, P6), (P4, P3), (P10, P7), (P12, P7), (Ciudad, P7), (P12, Ciudad), (D3, P2), (D1, P2), (D2, P2), (D4, P2), (P1, D3), (P1, D1), (P1, D2), (P1, D4), (P5, P2), (P4, P2)\}$$

Y por último resolviendo este problema de asignación obtenemos $z_{1.1.2.1.1.2.2.3.1.4.2.2} = 290$ con la siguiente asignación

$$D1 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow D4 \rightarrow P1 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow D3 \rightarrow P7 \rightarrow P10 \rightarrow P11 \rightarrow P12 \rightarrow D1$$

$$D2 \rightarrow P5 \rightarrow D2$$

Solución factible ya que no contiene ningún subtour ilegal, y como sabemos por el artículo de Dantzig y Ramser solución óptima. Cabe señalar que este problema tiene solución múltiple, ya que la asignación

$$D1 \rightarrow P6 \rightarrow P8 \rightarrow P9 \rightarrow D4 \rightarrow P1 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow D3 \rightarrow P7 \rightarrow P12 \rightarrow P11 \rightarrow P10 \rightarrow D1$$

$$D2 \rightarrow P5 \rightarrow D2$$

también es válida y su función objetivo tiene el mismo valor (290).